TCP/IP BOOT-PROM
User's Manual
Version 1.52

TCP/IP BOOT-PROM User's Manual

This manual was written, illustrated, and produced using FrameMaker publishing software on a NeXT workstation. Proofs were rendered by a NeXT computer using the Dots software and printed on a 600 dpi printer. Final pages were transferred directly from a NeXT disk to film.

Project team: Malcolm Agnew, Ralf Büttner, Dirk Köppen

Printed in Germany.

22.Dec 1993, covers TCP/IP BOOT-PROM software version 1.50.
10.Nov'1995, updated to cover TCP/IP BOOT-PROM software version 1.52.

Contents

Preface

This *User's Manual* explains how to:

- Install the TCP/IP BOOT-PROM.
- Configure the diskless clients and the server site.
- Create and maintain boot configurations.
- Use the TCP/IP BOOT-PROM *Utility Programs.*
- Trouble shoot problems.

It does not describe how to install and configure the network controller. See the network controller manual for this information. The TCP/IP BOOT-PROM's programming interface is also not covered by this manual.

Typographic Conventions

This book uses the following typographic conventions:

- *This font* is used for: emphasis; a command argument for which you must replace the argument name; or the title of a book. For example: *User's Manual*
- `This font` indicates text that the system displays on the screen. For example:
  `Press <SPACE> to abort!`
  It also indicates a filename or program.
- `This font` indicates what is typed by the user, as illustrated in screen examples:

```
C:\> format a: /u/s
```

Related Information

The following books and documentation contain more information on topics covered by this book:

- The TCP/IP BOOT-PROM *Product Data Sheet.*
- The paper *Introduction to the TCP/IP BOOT-PROM.*
- The TCP/IP BOOT-PROM *Technical Reference Manual..*
- Various *Request For Comments* from the INTERNET.
- IBM: Technical Reference Personal Computer AT, International Business Corporation, Boca Raton, Florida U.S.A., March 1984.
- Phoenix Technologies Ltd.: *System BIOS for IBM PCs, Compatibles and EISA Computers*, Addision-Wesley Publishing Company Inc., ISBN 0-201-57760-7, May 1991.
- Ray Duncan: *The MS-DOS Encyclopedia*, Microsoft Press, ISBN 1-55615-174-8, 1988

How to reach support

Updates and documentation can be obtained from the TCP/IP BOOT-PROM *BBS*:

- Via modem at +49 69 9894 1025.
- Via ftp from ftp.incom.dunkel.de.
- Via gopher from gopher.incom.dunkel.de
- Via WWW from http://www.dunkel.de/incom/

Support can be reached via electronic mail on support@incom.de.

Note - The number of the BBS service may disappear or change without notice. Refer to the file `SUPPORT.DOC` on the TCP/IP BOOT-PROM *Utility Disk*.

Introducing the TCP/IP BOOT-PROM

- The TCP/IP BOOT-PROM permits diskless Personal Computers to be booted from a network server.
- Various operating systems and network software is supported.
- The operating system and network software are held on a network server.
- The TCP/IP BOOT-PROM needs an IBM PC compatible Personal Computer with at least 640K byte RAM.
- After a diskless bootstrap the system works from a mounted network drive. Optionally installed diskette or hard disk drives are still supported and can be read/write protected.
- The TCP/IP BOOT-PROM supports the protocols DHCP, BOOTP and TFTP.

What BOOT PROMs are

When the IBM PC System BIOS was built, networking for Personal Computers was not very common and no support for network controllers was built into the System BIOS. A procedure within the System BIOS allows program code to be added to the System BIOS and stored in ROM. These ROMs are placed in a socket on the controller and contain hard-disk or network controller support which is processed by the System BIOS when the PC is powered on.

BOOT ROMs use the optional ROM feature and can start a PC from a network instead of the built in hard or floppy disk. There are two common ways to start PCs from a network, the first one is called *diskless* which means that no hard disk is present. The BOOT ROM retrieves the operating system, network and application software from the network.

The second is called *dataless*, which means that the PC is started from a built in hard disk which holds the operating system, network and application software. The data is stored on a network server for data security. In the latter case the BOOT ROM is used in a backup manner if the hard disk is damaged or its files are corrupted. The BOOT ROM then allows a diskless operation or can restore the data on the hard disk from the network server.

Advantages of BOOT-PROMs

There are several reasons for using diskless or dataless systems.

Security

- No floppy drives or hard disks are installed in the client PC; file access is controlled by the network software.

Maintenance

- No local hard disks to administer.
- Easy maintenance of configuration files on the server.
- Network servers can handle software licensing and access.
- Workstations can easily be replaced.

Economic

- Cost savings since either no or reduced numbers of hard and floppy disks are required.

Ergonomic

- Noise reduction due to no hard disk.
- Hard disk power saving can lead to fanless power supplies.

The TCP/IP BOOT-PROM

The product TCP/IP BOOT-PROM is a Boot ROM which uses the DHCP, BOOTP and TFTP protocols to implement a diskless or dataless boot of a PC. All protocols are common on TCP/IP networks and are defined in RFC-1531, RFC-951 and RFC-783. RFCs define protocol standards which are governed by the Internet.

The DHCP and BOOTP protocols allow network clients to determine their network configuration, address and boot host. Additions to the DHCP and BOOTP protocols allow the setting of user defined variables which can be used to configure a client, e.g. load site specific drivers or applications.

The network server runs a program known as the BOOTP daemon (`/etc/bootpd` on UNIX) which implements the server part of the BOOTP protocol. The BOOTP daemon configuration file (`/etc/-bootptab` on UNIX) holds a record for each client specifying its configuration. Configurations for similar clients can be grouped to simplify administration.

The DHCP protocol is an extension to the BOOTP protocol and allows dynamic host address configuration.

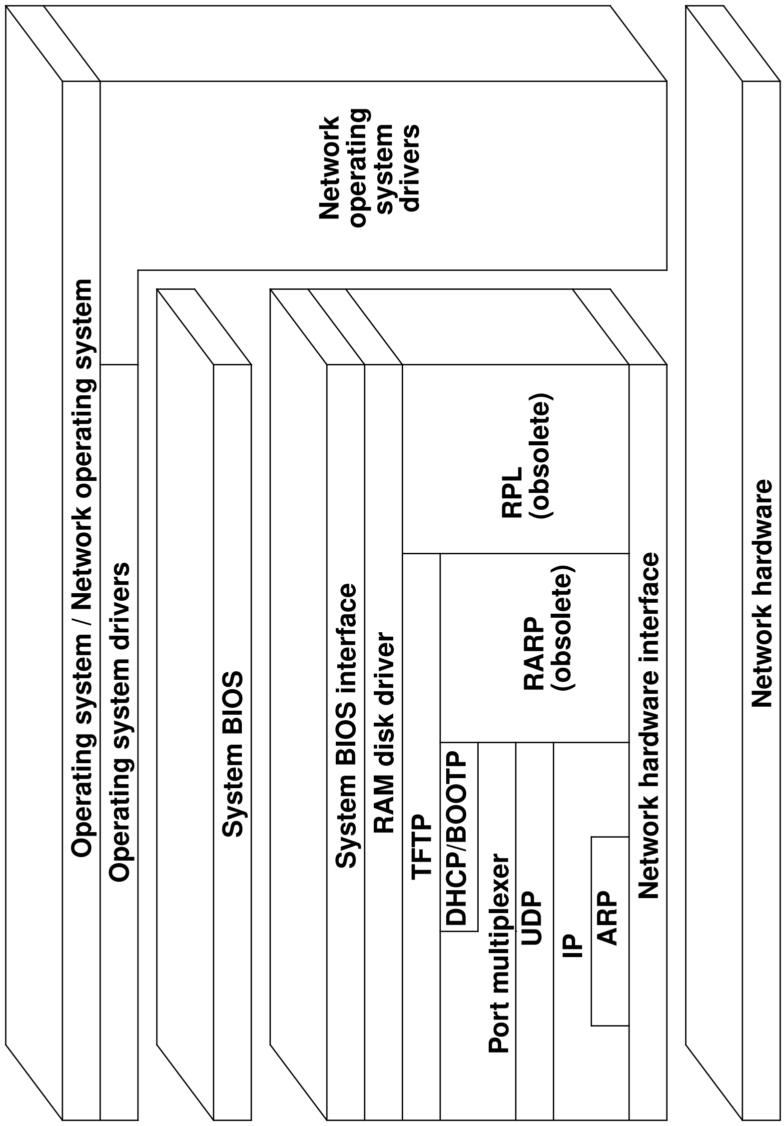
The TFTP protocol is a simple protocol to transfer a file from a server to a client or reverse. The TCP/IP BOOT-PROM uses the TFTP protocol to download the operating system and network software from the server. The client configuration and servername are obtained by the DHCP or BOOTP reply information from the initial inquiry as described above.

The DHCP, BOOTP and TFTP protocols are operating system independent. Servers are available for many operating systems including UNIX, NetWare, OS/2, VMS, MS-DOS and Windows WfW/95/NT. It is essential that the server system is based on the TCP/IP network protocol. If the server system does not implement the DHCP, BOOTP or TFTP protocols then public domain sources are available on the Internet.

Note - Not all versions of the TCP/IP BOOT-PROM implement the DHCP protocol. See chapter “Supported network controllers” on page 143 for detailed information.

How the TCP/IP BOOT-PROM works

After the ROM has been installed on the network controller, it sends out a combined DHCP and BOOTP request to inquire its configuration. If a valid reply is received, the ROM’s code starts a TFTP service to transfer the boot image containing the operating system (e.g. MS-DOS) and network software from the server to the client. The programs are stored in a single file called the *RAM disk image* or *container file* on the server. After the RAM disk image has been successfully transferred, the ROM installs this file as a RAM disk and executes a System BIOS bootstrap from that RAM disk.



The operating system then starts from the RAM disk and installs the network software which gives the PC a network drive. Further program loading is done from the network drive. The memory allocated by the RAM disk is freed by the ROM after a successful network logon. No resources are lost or allocated after a bootstrap using the TCP/IP BOOT-PROM. If the PC is rebooted, the ROM again gains control and starts the system from the network.

The network server configuration

The administrator creates a bootable diskette which holds the operating system, network software and additional programs which are needed at system start. This diskette can be tested by booting a PC from the diskette drive. If the diskette is known to work, an image is made from this diskette. This image (RAM disk image) can be created using an MS-DOS program which comes with the TCP/IP BOOT-PROM distribution or by using the `dd` program on UNIX systems.

The RAM disk image is transferred to the network server and a record in the DHCP or BOOTP configuration file (`/etc/bootptab`) is created for the client. This entry holds the network name, hardware address (MAC address), network address (IP address), name and location of the RAM disk image as well as site specific variables and additional information like routers. Several clients can share one RAM disk image to ease administration. The administrator must also enable the user of the client to use network services on the server. After this the network configuration is ready to support a diskless boot from the server.

Features of the TCP/IP BOOT-PROM

- An IBM PC AT or 100% compatible computer with at least 640K Byte of RAM is needed. Optional Extended Memory allows a larger RAM disk image to be loaded.
- The RAM disk image size can lie between 160K and 2.88 MB.
- In addition to the BOOTP and TFTP protocols some TCP/IP BOOT-PROMs also includes the DHCP protocol.
- Optionally installed diskette or hard disks are supported. Diskette drives can be read /write protected by password.
- Diskette or hard disk boots can be prevented. Network boot can be enabled by keystroke for dataless support.
- Built in controller and network diagnostics.
- Larger TFTP blocks for better performance.
- Multicast TFTP transfer for less network load.
- All ROMs are updateable remotely .

- A software updateable FLASH version is also available.
- Application interface (API) allows diskless clients to use the ROMs internal TCP/IP stack.
- Support of network routers and subnetting is provided.

The TCP/IP BOOT-PROM programs

BPSHELL

A window based menu MS-DOS program to:

- Create, restore and maintain RAM disk image files.
- Create and maintain BOOTP configuration files.
- Create passwords for diskette drive protection.
- Change parameters of the ROM image.
- Create and view menus by the `BPMENU` system.
- Read TCP/IP BOOT-PROM documentation.

BPUTIL

A MS-DOS program to:

- View and patch DHCP/BOOTP information such as network name, network address, serial numbers, site specific configurations and routing information into ASCII, binary and batch files.
- Protect the RAM disk image from being overwritten.
- Free the resources allocated by the TCP/IP BOOT-PROM.
- Transfer files using the TFTP protocol.

BPDISK

A MS-DOS (also available for the UNIX operating system) program to:

- Create and maintain RAM disk image files without a floppy drive.
- Create and maintain 2.88 MB RAM disk image files on 1.44 MB floppy disk equiped systems.
- Create and maintain RAM disk image files out of shell scripts or MS-DOS batch files.

BPMENU

A menu system to:

- Give the user the ability to choose a RAM disk image to be downloaded.
- Create a logon screen on the network client.
- Set default configurations and help screens.
- Create scripts by ASCII editor or the `BPSHELL` program. `BPSHELL` also allows the execution of scripts on MS-DOS systems for debugging.
- Handle several configuration by selection. For example a user can choose wether to boot diskless NFS, NetWare, LanManager or UNIX if the administrator provides RAM disk images for these applications.

BPROML

A program to download and execute ROM code from the network server. All TCP/IP BOOT-PROMs can be updated remotely using the `BPROML` program.

- Third-party NetWare, MOP[1] or LanManager ROMs can be downloaded.
- Allows secondary bootstrap procedures.

BPDLOAD

A program to load a ROM image from a diskette for systems where no socket for a PROM device is available like on PCMCIA adapters.

BPFLASH

A program to load the FLASH[2] of a network controller with the TCP/IP BOOT-PROM code.

The TCP/IP BOOT-PROM *Utility Program Diskette* also includes the source of the BOOTP/TFTP daemons, Novell NetWare 3.1x/4.x NLMs, OS/2 binaries and Windows applications for BOOTP and TFTP servers.

1. MOP is a Digital Equipment prorietary protocol to diskless boot computers.
2. FLASH memory is a writeable but non-volatile memory device.

About RARP and RPL

RARP and RPL are protocols with which a diskless Personal Computer can be booted. These protocols are not supported by the standard TCP/IP BOOT-PROM which uses DHCP and BOOTP.

A version of the TCP/IP BOOT-PROM called TCP/IP BOOT-PROM *Plus* implements RARP and RPL in addition to the DHCP and BOOTP protocol. The TCP/IP BOOT-PROM *Plus* is not available for all network controllers supported by the standard TCP/IP BOOT-PROM.

Note - The TCP/IP BOOT-PROM *Plus* is no longer available.

Installing the TCP/IP BOOT-PROM

Various brands of Ethernet and Token Ring controllers are supported by the TCP/IP BOOT-PROM. Since each network controller differs in the way it is detected, initialized and used, one specific driver for each particular board is resident in the ROM. A new TCP/IP BOOT-PROM must be purchased if the type of network controller hardware is changed.

The type of driver used can be detected by its name, e.g. the driver for the 3Com Etherlink II is named C503. When the ROM comes up it displays a message like the following:

```
TCP/IP BOOT-PROM 1.50, C503 1.10
(c) 1989,1993 Dirk Köppen EDV-Beratungs-GmbH, D-63073 Offenbach
Press <SPACE> to abort...
```

This shows that the version of the TCP/IP BOOT-PROM is `1.50` and that the type of network controller driver is a C503 with the version `1.10`.

The ROM might also display the following:

```
TCP/IP BOOT-PROM 1.52, C509 1.53MPDC
(c) 1989,1995 Dirk Köppen EDV-Beratungs-GmbH, D-63073 Offenbach
Press <SPACE> to abort...
```

This shows that the version of the TCP/IP BOOT-PROM is `1.52` and that the type of network controller driver is a C509 with the version `1.53`. The letters following the driver version point out that the PROM supports multicast TFTP (`M`), PCI local bus (`P`), DHCP (`D`) and PCMCIA bus (`C`).

Installing the BOOT ROM

The ROM device of the TCP/IP BOOT-PROM usually fits into a socket on the network controller but some motherboards have an extra socket in which an optional ROM can be placed or load the program code into a FLASH memory. Follow the instructions of your network controller for the correct insertion of the ROM device.

Note - If you insert the ROM device take care that the notch points in the right direction. The device will be damaged immediately after the PC is powered on if the device is inserted in the wrong direction.

In addition read the section in the following chapter "Notes on supported network controllers" which describes your network controller.

Note - If you use a controller with a bootable on-board BIOS or optional ROM (e.g. a SCSI or RLL controller) you must set this ROM's address to a value less than that of the TCP/IP BOOT-PROM. Otherwise the TCP/IP BOOT-PROM's bootstrap loader will not be able to install the controllers' BIOS ROM for the secondary boot stage. It is recommended to set the BIOS ROM address to `0xc8000` and the TCP/IP BOOT-PROM address to `0xcc000`.

How the ROM finds the network controller

All network controllers on EISA, MCA, PCI and PCMCIA systems can be auto detected. That is: the TCP/IP BOOT-PROM searches in all slots for the network controllers identification number and takes the first controller it finds.
Slots are searched in ascending order e.g. `0`, `1`, `2`, `3`.

On ISA systems not all network controllers can be auto detected. The TCP/IP BOOT-PROM then examines up to 5 addresses to look for the network controller. These addresses are hardwired into the ROM's program code but can be changed by using the `BPSHELL` program.

The type of detection algorithm, valid i/o addresses, tested configurations and features of your network controller can be looked up in the appendix "Supported network controllers" on page 143.

Notes on supported network controllers

3Com Etherlink II, Etherlink II-16 (C503)

- The ROM is installed in the socket on the network controller and enabled by setting the address jumper.
- The 3C503 will fail with PC-NFS if the memory address jumper is set. This can be solved by a switch to the PC-NFS device driver. According to your memory jumper setting, add the switch `/M:x` to your

`CONFIG.SYS` file:

```
device=a:\nfs\pcnfs.sys /m
device=a:\nfs\sockdrv.sys
device=a:\nfs\inivec.sys /m:x
device=a:\nfs\nfsvec.sys
```

where **x** is a number that depends on the position of your memory address jumper:

memory address	software switch
`memory disable`	`n/a`
`0xdc000`	`/M:1`
`0xd8000`	`/M:2`
`0xcc000`	`/M:3`
`0xc8000`	`/M:4`

- Personal Computers using the 3C503 cannot be rebooted via `<Ctrl>-<Alt>-<Del>` since the 3C503 replaces the installed ROM with its shared memory. The PROM is only re-enabled after a RESET or power down.
- It is not possible to use a BOOT-PROM in the 16 bit version of the 3C503 (Etherlink II/16 or 3C503-16) when the card is configured to 16 bit mode. You must setup the board to 8 Bit mode.
- By default the driver uses the BNC connector of the 3C503. To use the AUI connector, change the setting by using the `BPSHELL` program and reburn the ROM.

3Com Etherlink Plus (C505)

- The TCP/IP BOOT-PROM does not use the 3C505 with DMA transfer. Therefore the DMA setting is ignored by the TCP/IP BOOT-PROM.
- Error messages must be looked up in the table “Error messages” on page 149

3Com Etherlink 16 (C507)

- The 3C507 driver detects the RAM base address and size installed (16, 32, 48 or 64K). The RAM base address must lie within the first 1 M byte area.

3Com Etherlink III (C509)

- All settings of the 3C509 are auto detected. The 3C509 can be used in ISA and EISA mode.
- In EISA systems the ROM can also be used with the 3C579 network controller. Due to a bug in former EISA configuration files, the 3C579 may not work with the ROM set to address `0xcc000`.
- The ROM does not support the media type AUTO wich can be set using the 3Com configuration program. When AUTO is configured, the ROM uses the Twisted Pair port, e.g. 10Base-T.
- On the 3C509B disable the Plug and Play mode.

3Com Etherlink MC (C523)

- The ROM auto detects the card in any MCA channel.
- The address of the ROM and shared memory must lie in the first 1 M byte area.

3Com Etherlink III (C590)

- The ROM does not support the media type AUTO wich can be set using the 3Com configuration program. When AUTO is configured, the ROM uses the Twisted Pair port, e.g. 10Base-T.

3Com Etherlink III (C589)

- In PCMCIA systems which use an Intel SL 82365 B or compatible PCMCIA controller, the ROM scans the PCMCIA slots for a 3C589 or 3C589B first. Only Intel SL 82365 B or compatible PCMCIA controllers are supported.
- The ROM does not support the media type AUTO wich can be set using the 3Com configuration program. When AUTO is configured, the ROM uses the Twisted Pair port, e.g. 10Base-T.

3Com Etherlink III (C595)

- The ROM does not support the media type AUTO wich can be set using the 3Com configuration program. When AUTO is configured, the ROM uses the 100Base-TX mode.

3Com Etherlink III (C597)

- The ROM does not support the media type AUTO wich can be set using the 3Com configuration program. When AUTO is configured, the ROM uses the 100Base-TX mode.

Cabletron DNI E11xx, E21xx (CTRON21)

- The ROM auto-detects the size of memory installed and the type of DNI controller (E11xx or E21xx). It also supports the PAXI and PAXI+ version of the Cabletron chipset.
- If the adapter has multiple ports, the Twisted Pair port is tested for a valid link signal first. If no link can be detected the BNC/AUI/Fiber port is used.
- The ROM jumper block is set as follows:

JP2 jumpers	ROM address
`none`	`disabled`
`1-2,3-4`	`C000`
`3-4`	`D000`
`1-2`	`E000`

EDEN ED586 (ED586)

Since the ED586 has no on board ROM socket the ROM must be placed on the motherboard or a additional ROM carrier card.

ICL EtherTeam 16i (16ETHI)

- The driver uses the hardwired addresses to search for the network controller.
- The EEPROM is used to select the network port. If auto is set, the driver tries to find the connect port by sending out test packages.
- The ROM includes the XAPI header.
- On the ErgoPRO C4 workstation the ROM code must be enabled using the setup program.

Digital EtherWORKS LC, Turbo (EWORKS2)

The EWORKS2 driver supports the DE100 and DE200 Ethernet controllers. It autodetects the memory address and i/o address. Some differences between the Digital Equipment remote boot feature and the TCP/IP BOOT-

PROM are listed below:

- The diagnostics LED is turned on when the BOOT-PROM initializes the controller. It is turned off if the initialization was successful. The LED will stay turned on if the BOOT-PROM was not able to initialize the card, additionally an error will be logged to the screen.
- Each read or write to the network that is done by the BOOT-PROM will flash the LED.
- The driver supports small and big buffer mode. It distinguishes the installed mode automatically.
- The `RBE` (Remote Boot Enable) jumper can be used to turn off the TCP/IP BOOT-PROM. If `RBE` is jumpered so that remote boot is disabled, the BOOT-PROM notifies this by a message on the screen and continues with a diskette or hard disk bootstrap.
- The `TO` (Remote Boot Period Selection) jumper works differently to the one described in the EtherWORKS manual. If `TO` is jumpered to a 30 second time-out period, the BOOT-PROM will time-out after 60 seconds if it was not able to download a file. After the time-out, the BOOT-PROM will continue with a diskette or hard disk bootstrap.
  If `TO` is jumpered to 150 seconds, the BOOT-PROM also times out after 60 seconds but will reset the system and reboot cyclicly. No diskette or hard disk boot will be done in 150 second mode, the BOOT-PROM will try to bootstrap from the network infinitely.
- If a network driver disables the BOOT-PROM on the EtherWORKS card, the system will not restart from the BOOT-PROM after a `<Ctrl>-<Alt>-<Del>` sequence. In this case a RESET or power off/on cycle must be used to re-enable the BOOT-PROM.

Digital EtherWORKS III (EWORKS3)

The EWORKS3 driver supports the DE203, DE204 and DE205 Ethernet controllers. It autodetects the memory address and i/o address. If multiple Ethernet controllers are installed it uses the network controller on which the the ROM is mounted. Some differences between the Digital Equipment remote boot feature and the TCP/IP BOOT-PROM are listed below:

- The diagnostics LED is turned on when the BOOT-PROM initializes the controller. It is turned off if the initialization was successful. The LED will stay turned on if the BOOT-PROM was not able to initialize the card, additionally an error will be logged to the screen.

- Each read or write to the network done by the BOOT-PROM will flash the LED.
- The driver supports 2K, 16K and 32K mode. It distinguishes the installed mode automatically.
- The `NICSETUP` program must be used to enable the remote boot feature.
- If the network port has been set by `NICSETUP`, then this port is used, otherwise it auto detects the port connected to the network.

EXOS 105 (EXOS105)

It is not possible to detect the DMA jumper setting by software on this Ethernet controller. Therefore the PROM attaches the corressponding DMA channel to the i/o port which can be detected. That is, if you use i/o port `0x300` you must jumper the DMA channel to `DMA 5`. The following table gives an overview of all possible combinations that confirm to the settings specified by Novell:

i/o port	DMA channel
0x300	5
0x320	6
0x340	7
0x360	3

EXOS 205 (EXOS205)

The EXOS205 driver supports the EXOS 205T which can have 256K or 512K bytes of RAM installed.

Intel EtherExpress 16 (EXP16)

The EtherExpress driver detects the controller by its i/o port address and uses the controller via i/o mapped access. Therefore no shared memory is needed for this driver. See the enclosed table for all possible i/o port settings.

The Intel `SOFTSET` program (Version 1.12) does not support the 16K ROM used by the TCP/IP BOOT-PROM. Only 8K byte ROMs are supported. First check if there is a new version of `SOFTSET` available that supports 16K byte BOOT-ROMs. Otherwise a binary patch program is required to change the configuration to set the EtherExpress card into 16K ROM mode. This is how to use the program.

- Insert the ROM into the EtherExpress ROM socket.
- Turn on the PC and start `SOFTSET`.
- Use **`Manual setup`** to configure the EtherExpress.
- Configure the card as needed, take care that the i/o port is supported by the TCP/IP BOOT-PROM.
- Use **`C800-C9FFh`** for the `Boot ROM address`.
- Accept the configuration by pressing key **`<F10>`**.
- **`Save configuration`** to a file and exit `SOFTSET`.
- Use a binary patch program such as Norton `NU` to clear bit number `5` at file offset `0x19` (hexadecimal). For example if the bytes from offset `0x18` look like `00 2F 9D`, change them to `00 0F 9D`.
- Again start `SOFTSET` and **`Load configuration`** using the last file.
- Accept the configuration by pressing **`<SPACE>`**.
- Exit `SOFTSET` and turn your PC off and then on again.
- The TCP/IP BOOT-PROM should now come up with the copyright message.

Intel EtherExpress 16 FLASH (EXP16F)

The EtherExpress driver detects the controller by its i/o port address and uses the controller via i/o mapped access. Therefore no shared memory is needed for this driver. See the enclosed table for all possible i/o port settings.

- The EXP16F ROM cannot be used in the EtherExpress 16 (non-FLASH) board.
- The program code is stored in a file which is then loaded into the FLASH using the `BPFLASH` program. See chapter “Using BPROML” on page 89 on how to use the `BPFLASH` program.
- The program code can also be burned into a ROM device and be used instead of the FLASH.

ISOLAN 4102 (ISO4102)

It is not possible to detect the DMA jumper setting via software on the ISOLAN 4102 Ethernet controller. Therefore the PROM attaches the corressponding DMA channel to the i/o port which can be detected. That is, if you use i/o port `0x280` you must jumper the DMA channel to `DMA 0`. The following table gives an overview of all possible combinations which confirm to the settings specified by BICC:

i/o port	DMA channel
0x280	0
0x250	1
0x390	3
0x330	5
0x360	6

Novell NE1000 (NE1000)

- Since the NE1000 can only hold an 8K byte ROM, the error messages must be looked up in the enclosed table “Error messages” on page 149.
- The driver strips off the first byte of the Ethernet address if the network controllers address incorrectly sets the multicast address.

Novell NE2000 (NE2000)

The driver strips off the first byte of the Ethernet address if the network controllers address incorrectly sets the multicast address.

Novell NE2000 Plus (NE2000P)

- Since the NE2000 Plus can only hold an 8K byte ROM, the error messages must be looked up in the enclosed table “Error messages” on page 149.
- The driver strips off the first byte of the Ethernet address if the network controllers address incorrectly sets the multicast address.
- The controller must be configured for i/o mapped mode.

Novell NE2100 (NE2100)

The NE2100 is compatible to the EXOS 105. See the section “EXOS 105 (EXOS105)” on page 27.

INTERLAN NI5210 (NI5210)

- The 8K byte ROM replaces one RAM on the NI5210.
- The error messages must be looked up in the enclosed table “Error messages” on page 149.

INTERLAN NI6510 (NI6510)

The driver searches for the network controller at the four possible i/o port settings of the NI6510. It then reads the jumpered settings.

INTERLAN NI9210 (NI9210)

- The card is detected by the MCA card ID.
- The driver uses the card in shared memory mode, all sizes are supported.
- The addresses of the ROM and the shared memory must be within the first 1M byte.

Packet Driver version (PKTDRV)

- The program code searches for the packet driver at the hardwired addresses.
- The code is intended to be run with a specialized application. It cannot be used stand-alone.

Pure Data PD 8023 Plus (PD8023)

- Since the PD8023 can only hold 8K byte ROMs the error messages must be looked up in the enclosed table “Error messages” on page 149.
- The card is searched at the hardwired addresses of the TCP/IP BOOT-PROM. All other configurations are read out of the cards EEPROM setting.

Schneider & Koch SK-NET junior (SKJUN)

The card does not have a ROM socket therefore the ROM must be placed on an add-on card.

Schneider & Koch SK-NET G16 (SKG16)

The driver uses the G16 shared memory immediately after the ROM address.

Schneider & Koch SK-NET MC2 (SKMC2)

The driver auto detects the card in all MCA slots.

Schneider & Koch SK-Y-NET (SKYNET)

The SK-Y-NET is compatible to the EXOS105. See the section “EXOS 105 (EXOS105)” on page 27.

SMC EtherCard Elite, Ultra, EtherEZ (SMC8416)

- The ROM is a 32K bytes device. The space allocated by the ROM can be used my `EMM386` or `QEMM`.
- The driver supports all SMC and Western Digital EtherCard in ISA and MCA slots.
- The setting of the card are read out of the EEPROM.
- The former SMCETH ROM has been replaced by the SMC8416 ROM.

SMC TokenCard Elite (SMC8115)

- The TokenCard driver uses the SNAP protocol to encapsulate the IP protocol into the Token Ring frames.
- You can use `EMM386` or `QEMM` to use the 64K byte range of ROM space that the PROM allocates for Upper Memory by including that range.
- The former SMCTOK ROM has been replaced by the SMC8115 ROM.

Western Digital EtherCard Elite (WD8ISA)

- The driver ignores the EEPROM configuration and uses its own table to be compatible to older and compatible network controllers.
- If the device driver is not able to find a valid address at which to place the shared memory, it uses the segment `0xd000` by default.

Western Digital EtherCard Elite/A (WD8MCA)

The driver detects the card in the MCA slots and uses the configured settings of the controller.

Testing the TCP/IP BOOT-PROM

After the installation of the ROM the Personal Computer will come up executing the System BIOS self-test and then display the TCP/IP BOOT-PROM copyright message:

```
TCP/IP BOOT-PROM 1.50, WD8ISA 1.10
(c) 1989,1993 Dirk Köppen EDV-Beratungs-GmbH, D-63073 Offenbach
Press <SPACE> to abort...
```

If you do not get this message the ROM is not installed correctly. Several reasons may cause this problem:

- The ROM is disabled by the network controller configuration.
- An incorrect ROM size is configured on the controller.
- The ROM area overlaps with a shared memory or other ROM.
- Another optional ROM breaks the interrupt chain. Remove all option ROMs for testing.
- The ROM is placed at an address outside the range `0xc800-0xdfff`.
- The ROM is damaged or has been inserted in the wrong direction.

Check also with the chapter “Troubleshooting and techniques” on page 137.

If **`<SPACE>`** is pressed within 3 seconds, the TCP/IP BOOT-PROM routines will be aborted and the System BIOS bootstrap loader invoked as usual.

Note - According to the configuration of your TCP/IP BOOT-PROM it may not be possible to boot from a diskette or hard disk drive after the network failed. Several options can be set by using the `BPSHELL` program.

If you do not press `<SPACE>` at this point, the TCP/IP BOOT-PROM routines will start and try to boot from the network. If there is no BOOTP server installed the TCP/IP BOOT-PROM ROM based software will time-out after a while and pass control to the System BIOS bootstrap loader.

Note - The TCP/IP BOOT-PROM may abort with an error message at this point. ROMs which can only hold 8K byte do not include error messages to save ROM space. These ROMs only show the error code, the text can be looked up in the chapter “Error messages” on page 149.

The possible keystrokes at the TCP/IP BOOT-PROM prompt are:

keypress	function
`<SPACE>`	Abort the TCP/IP BOOT-PROM and boot from the built in disk.
`d`	Enable diagnostic mode.
`v`	Enable verbose mode.

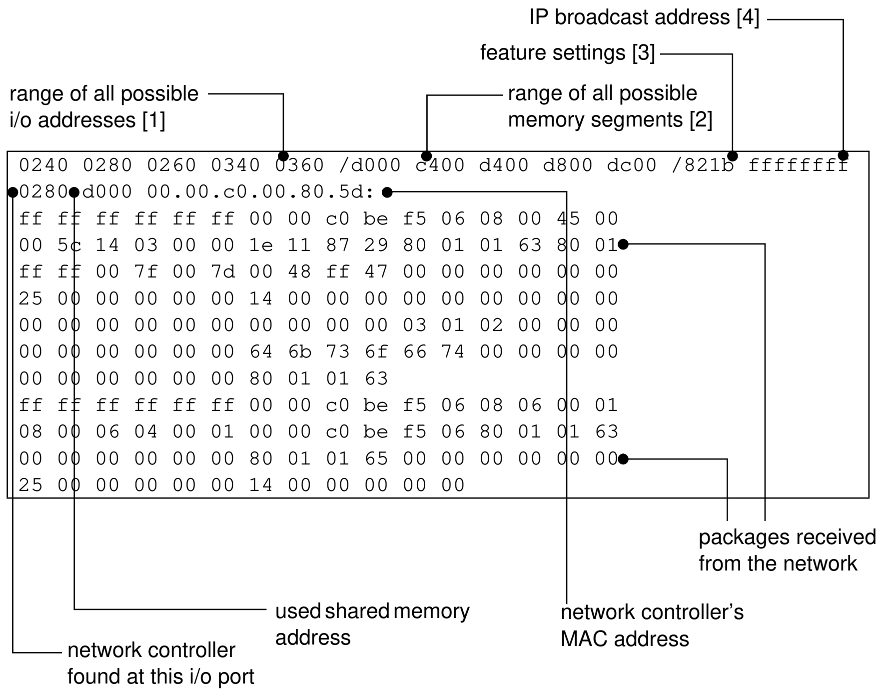
keypress	function
`a`	Force use of the RARP protocol. (TCP/IP BOOT-PROM *Plus* only and obsolete)
`b`	Force use of the BOOTP protocol. (TCP/IP BOOT-PROM *Plus* only and obsolete)
`r`	Force use of the RPL protocol. (TCP/IP BOOT-PROM *Plus* only and obsolete)

The TCP/IP BOOT-PROM diagnostic mode

If **<d>** is pressed while the TCP/IP BOOT-PROM displays `Press <SPACE> to abort..` the diagnostics mode is enabled and will be started after about 3 seconds.

Note - The diagnostic mode is useful to check whether the network controller can be initialized by the TCP/IP BOOT-PROM and establish the network controller's MAC address. Not all ROMs have the full diagnostic mode implemented. All 8K byte ROM versions only show the configuration and MAC address but do not execute a network trace.

In diagnostic mode the TCP/IP BOOT-PROM will find and initialize the network controller. The WD8ISA ROM displays a message similar to the following:



In detail the information is as follows:

[1] The range of i/o ports where the TCP/IP BOOT-PROM searches for the network controller. If no network controller is found at one of theses addresses it aborts with an error message.

[2] The range of segments the TCP/IP BOOT-PROM can use at which the the network controller's shared memory is allocated. On network controllers with DMA access this specifies the DMA channels that can be used.

[3] The enabled features of the ROM. Features can read/write protect diskette drives, lock the system or define the bootstrap devices. This coding can be set using the `BPSHELL` program.

[4] The IP broadcast address, using `BPSHELL` this can be set to `0` on older IP networks or to a subnet broadcast address.

After this the ROM starts receiving packages from the network. This shows that the TCP/IP BOOT-PROM is able to receive packages and the network controller is set up correctly. The type of packages received are always broadcast or packages with the destination address identical to the network controllers MAC address.

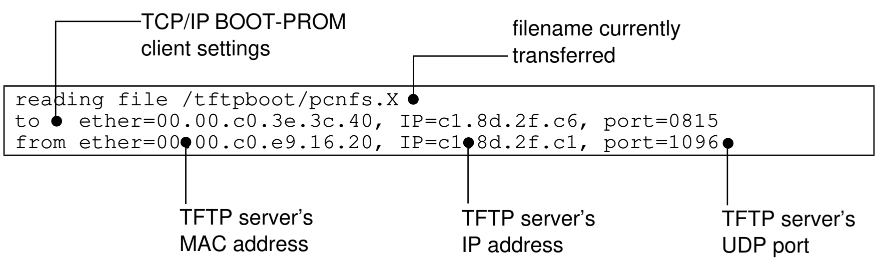
You can abort the diagnostic mode by pressing `<SPACE>`.

The TCP/IP BOOT-PROM verbose mode

If you press **<v>** while the TCP/IP BOOT-PROM displays `Press <SPACE> to abort..` the verbose mode is enabled and will start after about 3 seconds.

Note - The 8K byte versions of the TCP/IP BOOT-PROM do not implement the verbose mode.

The verbose mode gives more detailed information about the ROM activities. On Token Ring controllers the Ring Insert procedure can be monitored. By enabling the verbose mode the following output is displayed when the ROM downloads the RAM disk image via TFTP:



Note - The verbose mode is useful to find out the filename that the TCP/IP BOOT-PROM client will download from the network server. All numbers are hexadecimal.

ROMs that implement the DHCP protocol show some more specific information like:

```
DHCP: sip=c2.37.f7.c1, gip=00.00.00.00, yip=c2.37.f7.c6
DHCP: gw0=c2.37.f7.c1, gw1=00.00.00.00, smf=ff.ff.ff.f0
DHCP: bfn=/tftpboot/win95.PX
reading file /tftpboot/win95.PX
to   ether=00.00.c0.3e.3c.40, IP=c2.37.f7.c6, port=0815
from ether=00.00.c0.e9.16.20, IP=c2.37.f7.c1, port=1096
```

In addition to the previous output this adds the BOOTP or DHCP reply information:

tag	paramter
`sip`	The TFTP server's IP address.
`gip`	The DHCP/BOOTP forwarder's (gateway)[1] IP address.
`yip`	The client's IP address.
`gw0`	The first IP gateway's IP address.
`gw1`	The second (backup) IP gateway's IP address.
`smf`	The IP subnet mask.
`bfn`	The filename of the bootfile.

1. The DHCP/BOOTP gateway is a forwarder of the DHCP/BOOTP request package which may be different from the IP gateway.

Installing BOOTP and TFTP

Several BOOTP daemons are available. Check your manual for the correct installation. The following paragraphs show by example how to enable and configure the most common BOOTP daemons.

The full capabilities of the TCP/IP BOOT-PROM can only be used with the CMU[1] BOOTP daemon or any BOOTP daemon 2.4.0 or larger. To determine whether your system uses the CMU BOOTP daemon check with the manual page entry. The source code of a BOOTP daemon is also available on the TCP/IP BOOT-PROM *Utility Disk* or the TCP/IP BOOT-PROM BBS which can be compiled if no BOOTP daemon is available on the target system. BOOTP and TFTP for Novell NetWare and MS-Windows are available from then same source.

Configuring a CMU BOOTP daemon

This is a sample configuration using the CMU BOOTP daemon on an INTERACTIVE UNIX system.

Check in `/etc/services` that all the required ports for BOOTP and TFTP are listed. This file should contain the following entries:

```
bootps          67/udp          bootp server
bootpc          68/udp          bootp client
tftp            69/udp
```

1. CMU is the short form of Carnegie Mellon University

The next step is to start the BOOTP and TFTP daemons from the Internet daemon (`inetd`). This will be done automatically if the network client sends BOOTP or TFTP requests to the server. The file `/etc/inetd.conf` should hold entries like:

```
bootps   dgram    udp      wait     root     /etc/bootpd      bootpd
tftp     dgram    udp      wait     bin      /etc/tftpd       tftpd
```

If your `/etc/inetd.conf` file did not contain the above entries, restart the Internet daemon. This can be done by:

```
# ps -aef | grep inetd
  root    115     1  0  Aug 15  ?        0:01 /etc/inetd
  root   5463  5429  1 12:35:19 ttyp0    0:00 grep inetd
# kill -1 115
# _
```

After this, install the directories in which you want to store your RAM disk images. A common directory is `/tftpboot`:

```
# mkdir /tftpboot
# chmod a+rx /tftpboot
# _
```

All other configurations are done in the `/etc/bootptab` file. This configuration is site specific. A sample diskless client entry would be:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# ISDN Router
router:\
    :tc=ether:ha=9010020756BF:ip=193.141.47.199:\
    :hd=/tftpboot:bf=router.X:
```

Note - On the INTERACTIVE host based TCP/IP version 1.30 and 1.40, the BOOTP daemon may ignore the `hd` entry in `/etc/bootptab`. If so, give the absolute path name of the file for the `bf` entry, e.g. `bf=/tftpboot/router.X`.

Configuring a RFC 951 BOOTP daemon

This is a sample configuration using the RFC951 BOOTP daemon on a MACH operating system.

Note - The RFC 951 BOOTP daemon has a limited capability. The CMU BOOTP daemon is recommended in order to use all the configurations described in this manual.

Check in `/etc/services` that all ports needed for BOOTP and TFTP are listed. This file should contain the following entries:

```
bootps          67/udp          bootp server
bootpc          68/udp          bootp client
tftp            69/udp
```

The next step is to start the BOOTP and TFTP daemons from the Internet daemon (`inetd`). This will be done automatically if the network client sends BOOTP or TFTP requests to the server. The file `/etc/inetd.conf` should hold entries like:

```
bootps  dgram   udp     wait    root    /usr/etc/bootpd     bootpd
tftp    dgram   udp     wait    bin     /usr/etc/tftpd      tftpd
```

If your `/etc/inetd.conf` file did not contain the above entries you must restart the Internet daemon. This can be done by:

```
# ps -auxc | grep inetd
  root       135   0.0  0.9 1.52M  280K ?  SW    0:00 inetd
# kill -1 135
# _
```

After this, install the directories in which you want to store your RAM disk images. A common directory is `/tftpboot`:

```
# mkdir /tftpboot
# chmod a+rx /tftpboot
# _
```

All other configurations are done in the `/etc/bootptab` file. This configuration is site specific. A sample diskless client entry would be:

```
/tftpboot
mach
%%
diskless         1  0:0:c0:26:55:4e  193.141.47.198   ramd.X
```

Options of the TCP/IP BOOT-PROM BOOTP daemon

The BOOTP daemon on the TCP/IP BOOT-PROM *Utility Diskette* is a port of the CMU BOOTP daemon with advanced options. The following arguments can be given to the BOOTP daemon::

```
bootpd -i ipaddr[:sbits|/smask] [-d] [-s] [-t #] [cfgfile [dumpfile]]
```

Where:

argument	meaning
`-i ipaddr[/subnet]`	The IP address of the network interface that is to be used with BOOTP. You need to add the subnet mask if you are using subnetting.
`-i ipaddr[:sbits]`	Same as above but allows to specify the subnet by number of subnet bits instead of the subnet mask. You need to add the number of subnet bits if you are using subnetting.
`-d`	Increase debugging output. This option can be used up to four times.
`-s`	Enable stand-alone mode (always true).
`-t #`	Abort the BOOTP daemon after # minutes of inactivity. Use 0 to never time out.
`configfile`	Filename which holds the BOOTP daemon configuration (default: `/etc/bootptab`).
`dumpfile`	Filename to dump the BOOTP daemon configuration to (default: `/etc/bootpd.dump`).

Note - The TCP/IP BOOT-PROM BOOTP daemon can not be started using the Internet daemon (inetd).

If you have multiple network interfaces on which you want to enable BOOTP, use multiple `-i` options, e.g.:

```
bootpd -i 193.141.47.195 -i 193.141.48.1 -d -d -d -d -t 0 -s
```

If you use subnetting you must add the subnet mask to the `-i` option:

```
bootpd -i 193.141.47.195/255.255.255.240 -d -d -d -d -t 0 -s
```

If the command line is longer than your operating system is able to hold, you can write all arguments and optionally comments into a file, e.g. `/etc/bootpd.arg`:

```
# Ethernet interface
-i 193.141.47.195/255.255.255.240
# Token-Ring interface
-i 193.141.47.209/255.255.255.240
# other options
-d -d -d -d -t 0 -s
```

And pass them to the BOOTP daemon using the @ option:

```
bootpd @/etc/bootpd.arg
```

The BOOTP daemon from the TCP/IP BOOT-PROM *Utility Diskette* adds the following features:

- New tag ms to increase the BOOTP reply size.
- Automatic detection of maximum usable BOOTP reply size.
- New tag mw to delay a BOOTP reply.
- New tag sa to overwrite the boot server's IP address.

Increasing the BOOTP reply size

The standard BOOTP reply is limited to 300 bytes which includes 64 bytes of vendor specific information. Using the tag ms in /etc/bootptab you can increase the size of the BOOTP reply which allows to pass more vendor specific information with the BOOTP reply.

For example: the default BOOTP reply size is 328 bytes (300 bytes of BOOTP reply structure plus 28 bytes of UDP/IP information). By changing the BOOTP reply size to 776 bytes you get 512 bytes of vendor specific information. To increase the BOOTP reply size use the ms tag:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# sample entry
diskless:\
    :tc=ether:ha=0000c00756bf:ip=193.141.47.198:\
    :hd=/tftpboot:bf=ramd.X:ms=776:
```

Note - The tag ms is not needed in conjunction with the TCP/IP BOOT-PROM as the TCP/IP BOOT-PROM automatically sets the maximum size of the BOOTP reply in it's BOOTP request. The TCP/IP BOOT-PROM is able to receive BOOTP replies of up to 1288 bytes.

Delaying BOOTP replies

You can delay a BOOTP reply until the requesting client exceeds a threshold. If you set a threshold (seconds) using the tag mw, the BOOTP daemon starts sending BOOTP replies only after that limit is reached.

On this example the BOOTP server will wait until the client sends BOOTP requests which are stamped for 8 seconds of requesting time (time since the client started sending BOOTP requests). After that the BOOTP server will reply to the BOOTP request:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# sample entry
diskless:\
    :tc=ether:ha=0000c00756bf:ip=193.141.47.198:\
    :hd=/tftpboot:bf=ramd.X:mw=8:
```

This feature is useful if you have two BOOTP daemons running in one network where the second BOOTP daemon delays it's BOOTP replies. On this configuration the second BOOTP daemon only answers BOOTP replies if the first (undelayed) BOOTP daemon is down. So the second BOOTP server acts as a BOOTP backup server.

Changing the boot server's IP address

By default the BOOTP daemon uses it's own IP address as the server's IP address which may be used for a later TFTP transfer. You can overwrite this IP address using the tag `sa`.

This example sets the server 193.141.47.193 as the boot server from which the bootfile `/tftpboot/ramd.X` is transferred:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# sample entry
diskless:\
    :tc=ether:ha=0000c00756bf:ip=193.141.47.198:\
    :hd=/tftpboot:bf=ramd.X:sa=193.141.47.193:
```

Options of the TCP/IP BOOT-PROM TFTP daemon

The TFTP daemon on the TCP/IP BOOT-PROM *Utility Diskette* is a standalone program. It adds several features which are not available with the standard UNIX TFTP daemon (`tftpd`):

- Allows restricted file access for security.
- Does not spawn (fork) a new process on every TFTP transfer (less server load).
- Allows larger TFTP segment size for better performance and less network transactions.

- Supports multicast file transfer (MTFTP) for faster boot of concurrent booting diskless clients.
- Available for Windows, NetWare, OS/2 and UNIX.

Note - This TFTP daemon does not support writing files to the server.

The following arguments can be given to the TFTP daemon:

```
tftpd [-c num] [-d [path]] [-h] [-i num] [-k num] [-l [file]] \
      [-m file port] [-p num] [-r] [-s num port] [-u num] [-v 0-2]
```

Where:

argument	meaning
`-c num`	Number of concurrent TFTP transfers.
`-d [path]`	Change to directory before TFTP transfer.
`-h`	Enable read-ahead buffer.
`-i num`	Exit after num seconds of inactivity.
`-k num`	Set TFTP transmission keep-alive time.
`-l [file]`	Log all messages to file.
`-m file port`	MTFTP configuration file and listen port.
`-p num`	Set TFTP listen port.
`-r`	Restrict file access to current directory.
`-s num port`	Set TFTP segment size and listen port.
`-u num`	Number of MTFTP unicasts to be send.
`-v 0-2`	Set level of verbosity.

The -c option

The number of concurrent TFTP transfers can be limit using the `-c` option. The following example limits the number of concurrent TFTP transfers to `4`:

```
tftpd -c 4
```

The -d option

A default path can be added to each TFTP read request. The TFTP daemon does a chdir into the specified directory before it opens the requested file. If no directory is specified, the default `/tftpboot` (UNIX), `c:\tftpboot` (Windows) or `sys:\tftpboot` (NetWare) is choosen. The following examples load all relative named files out of the directory `/tftp-`

boot:

```
tftpd -d /tftpboot
tftpd -d
```

The -h option

This option enables read-ahead buffers for the TFTP file transfer. That is the TFTP daemon reads more data than needed and buffers the additional data for later use. In some environments the transfer speed increases by enabling the `-h` option. By default the TFTP daemon does not read-ahead.

The -i option

The TFTP daemon can be automatically terminated by using the `-i` option. If there is no TFTP activity for the number of seconds specified, the TFTP daemon is shut down. The following example aborts the TFTP daemon after 45 seconds of inactivity:

```
tftpd -i 45
```

The -k option

The TFTP daemons holds all information of a TFTP transmissions for a specified time so a client can re-enable an already finished transmission. The `-k` option allows to change the default time which is 15 seconds.

The -l option

All output of the TFTP daemon can also be logged into a filename. The `-l` option enables logging and allows to specify a filename. If no filename is passed to the `-l` option, the default `/tftpboot/tftpd.log` (UNIX), `c:\tftpboot\tftpd.log` (Windows) or `sys:\tftpboot\tftpd.log` (NetWare) is used. This example enables full debugging output and logs all output to `/tmp/tftpd.log`:

```
tftpd -v 2 -l /tmp/tftpd.log
```

The -m option

The `-m` option enables the TFTP daemon's multicast file transfer support (MTFTP). Two arguments are to be passed to the `-m` option. The first argument specifies a configuration file, the second argument specifies the UDP port number where the TFTP daemon should listen for MTFTP file transfer requests. The following example uses port number 75 for MTFTP requests:

```
tftpd -m /etc/mtftptab 75
```

The configuration file has the following format:

```
/tftpboot/pcnfs.X       225.0.0.1     76
/tftpboot/ramd.X        225.0.0.2     76
```

Note - As there are no reserved port numbers for the multicast TFTP it can not be guaranteed that the used ports 75 and 76 are operateable on all systems.

Where the first column holds the filename to be transmitted. If the TFTP daemon receives a MTFTP request for that filename on the specified MTFTP server port (in the above example 75), it sends the MTFTP packages to the IP address specified in the second column (225.0.0.1). The third column holds the MTFTP client port number where all data is send to.

The -p option

By default the TFTP daemon uses UDP port number 69 to listen for TFTP file transfer requests. This can be overridden by using the `-p` option. The TFTP daemon can also be used together with an existing TFTP daemon to use only the extended features of the TFTP daemon that comes on the TCP/IP BOOT-PROM *Utility Diskette*.

To disable the standard TFTP functionality of this TFTP daemon, bind the TFTP UDP port number to an unused UDP port, e.g.:

```
tftpd -s 1408 59 -m /etc/mtftptab 75 -v 1 -h -p 21435
```

The -r option

The `-r` option restricts all file access to the current directory. That is no other files than files out of the current directory or it's subdirectories can be transmitted. The current directory can be changed using the -`d` option.

The following example limits all file transmissions to the directory `/tftpboot` and it's subdirectories:

```
tftpd -d /tftpboot -r
```

The -s option

This option allows to set the TFTP segment size. The standard TFTP protocol specifies a segment size of 512 bytes. By increasing the segment size less network transactions are needed and therefore a faster file transfer can be achieved. Increasing the TFTP segment size makes the TFTP transmission incompatible with standard TFTP clients and the RFC specification. Therefore the TFTP daemon only sends larger packages to clients which open the TFTP transmission at the specified UDP port:

```
tftpd -s 1408 59
```

This adds support for larger TFTP packages (here 1408 bytes) if the connection is opened at UDP port 59 instead of the standard TFTP port 69. Opening a TFTP transmission at the standard TFTP port 69 let's the TFTP daemon send 512 bytes packages. Opening a TFTP transmission at the non-standard UDP port 59 will have the TFTP daemon sending 1408 bytes TFTP packages.

Note - If the bootfilename ends with the extension `P` or `PX` (e.g. `/tftpboot/ramd.PX`), the TCP/IP BOOT-PROM tries to open the TFTP transmission at port UDP 59 first. If this fails, the TCP/IP BOOT-PROM retries the TFTP transmission at standard TFTP port 69.

The -u option

The `-u` option is used in conjunction with the multicast file transfer (MTFTP). The `-u` option specifies the number of packages the TFTP daemon sends unicast to the MTFTP client after a transmission was opened.

The default is 4 packages and should not be changed.

The -v option

The debugging and activity output of the TFTP daemon can be changed by the `-v` option. The default is 1, increasing the number generates more output. 0 does not create any output.

Recommend usage of the TFTP daemon

The following usage can be used together with an existing TFTP daemon in order to add larger TFTP package support only, it also enables file access restriction:

```
tftpd -v 1 -s 1408 59 -r -d /tftpboot -p 21435
```

These arguments additionally replace the existing TFTP daemon:

```
tftpd -v 1 -s 1408 59 -r -d /tftpboot
```

To enable read-ahead and multicast file transfer use the following arguments:

```
tftpd -v 1 -s 1408 59 -r -d /tftpboot -h -m /etc/mtftptab 75
```

Configuring the BOOTP and TFTP daemons on NetWare

The Novell NetWare 3.1x operating system is not delivered with a BOOTP and TFTP daemon. You must install the daemons from the TCP/IP BOOT-PROM *Utility Diskette*.

The BOOTP daemon on the TCP/IP BOOT-PROM *Utility Diskette* is a port of the CMU BOOTP daemon with advanced options for the NetWare 3.1x and 4.x operating system.

To install, copy the files `TFTPD.NLM` and `BOOTPD.NLM` from the TCP/IP BOOT-PROM *Utility Diskette* to your network server's `sys:\system` directory.

Check in `sys:\etc\services` that all ports needed for BOOTP and TFTP are listed. If you do not have a `sys:\etc\services` file, copy the example out of the `sys:\etc\samples` directory to `sys:\etc\services`.

The file `sys:\etc\services` should contain the following entries:

```
bootps          67/udp          bootp server
bootpc          68/udp          bootp client
tftp            69/udp
```

Now edit the `sys:\system\autoexec.ncf` file to start the TFTP and BOOTP daemons at system start. A sample could be:

```
file server name nw
ipx internal net deadbeef

load smcnw311 port=240 int=3 mem=d4000 frame=ETHERNET_II name=en0
bind ipx to en0 net=1

load tcpip forward=yes rip=yes
bind ip to en0 addr=193.141.47.195

load tftpd -v 1 -r -d sys:\tftpboot -s 1408 59
load bootpd -i 193.141.47.195 -t 0 -d -d -d -d
```

Note - For further information on the options of the BOOTP and TFTP daemons see the enclosed chapter of the BOOTP and TFTP daemons.

Now create a directory `sys:\tftpboot` in which to store all RAM disk images. The directory and all RAM disk images should have read permission for all users.

The format of the `sys:\etc\bootptab` file is identical to the CMU BOOTP daemon. A sample entry would be:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# sample entry
diskless:\
    :tc=ether:ha=0000c00756bf:ip=193.141.47.198:\
    :hd="SYS:/TFTPBOOT":bf=ramd.X:
```

After this the system must be restarted to load the required daemons.

Configuring the BOOTP and TFTP daemons on Windows

The BOOTP and TFTP daemons from the TCP/IP BOOT-PROM *Utility Diskette* use the Winsock network interface and can therefore be used in the Microsoft Windows environment like on Windows 3.1 or Windows NT.

The BOOTP daemon on the TCP/IP BOOT-PROM *Utility Diskette* is a port of the CMU BOOTP daemon with advanced options for the Winsock network interface.

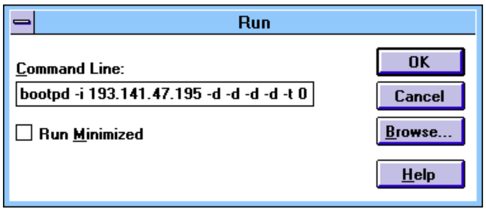
To install, copy the files `TFTPD.EXE` and `BOOTPD.EXE` from the TCP/IP BOOT-PROM *Utility Diskette* to your system's binary directory, e.g. `C:\ETC`.

Check in the `services` file that all ports needed for BOOTP and TFTP are listed. If you do not have a `services` file, the daemons will use their default values.

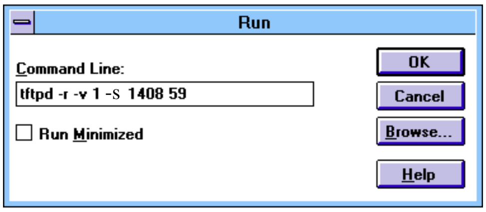
The file `services` should contain the following entries:

```
bootps          67/udp          bootp server
bootpc          68/udp          bootp client
tftp            69/udp
```

Then use the Filemanager to run the BOOTP daemon:

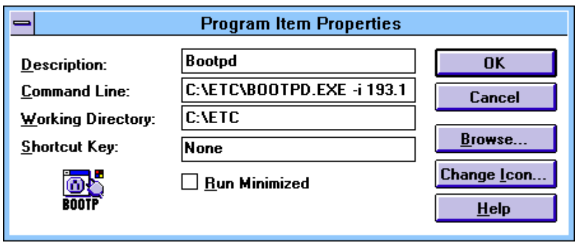


And run the TFTP daemon:



Note - For further information on the options of the BOOTP and TFTP daemons see the enclosed chapter of the BOOTP and TFTP daemons.

Instead of running the daemons via the Filemanager's Run menu, you can also create a Program Item Property:



Now create a directory `C:\TFTPBOOT` in which to store all RAM disk images. The directory and all RAM disk images should have read permission for all users if you run Windows NT.

The format of the `C:\ETC\BOOTPTAB` file is identical to the CMU BOOTP daemon. A sample entry would be:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# sample entry
diskless:\
    :tc=ether:ha=0000c00756bf:ip=193.141.47.198:\
    :hd=/TFTPBOOT:bf=ramd.X:
```

Configuring the TFTP daemon on UNIX

You can replace your existing TFTP daemon or add the TFTP daemon from the TCP/IP BOOT-PROM *Utility Diskette* to make use of the extended features of the TFTP daemon which comes on the TCP/IP BOOT-PROM *Utility Diskette*.

Note - The TCP/IP BOOT-PROM *Utility Diskette* may not hold a binary version of the TFTP daemon for your type of the UNIX operating system. Please check for availability with your vendor.

To install, copy the file `tftpd` from the TCP/IP BOOT-PROM *Utility Diskette* to your system's binary directory, e.g. `/etc`.

Check in the `/etc/services` file that all ports needed for the TFTP daemon are listed. The file `/etc/services` should contain the following

```
tftp            69/udp
```

entry:

If you want to replace the existing TFTP daemon remove it's entry from `/etc/inetd.conf` and restart your `inetd`. Then create a start-up script to start the TFTP daemon at each system start. You may want to put this script into `/etc/rc3.d`:

```
/etc/tftpd -v 0 -d /tftpboot -h -r -s 1408 59 &
```

If you do not want to replace the existing TFTP daemon do not remove it's entry out of `/etc/inetd.conf` and create the following script:

```
/etc/tftpd -v 0 -d /tftpboot -h -r -s 1408 59 -p 21435 &
```

Note - For further information on the options of the BOOTP and TFTP daemons see the enclosed chapter of the BOOTP and TFTP daemons.

Now create a directory `/tftpboot` in which to store all RAM disk images. The directory and all RAM disk images should have read permission for all users.

Configuring the BOOTP and TFTP daemons on OS/2

The BOOTP daemon on the TCP/IP BOOT-PROM *Utility Diskette* is a port of the CMU BOOTP daemon with advanced options for the OS/2 operating system. You need to install OS/2 TCP/IP in order to operate the daemons.

To install, copy the files `tftpd.exe` and `bootpd.exe` from the TCP/IP BOOT-PROM *Utility Diskette* to your OS/2 system's binary directory, e.g. `C:\bin` directory.

Check in `C:\tcpip\etc\services` that all ports needed for BOOTP and TFTP are listed. The file `C:\tcpip\etc\services` should contain the following entries:

```
sbootp          67/udp          # bootp server
cbootp          68/udp          # bootp client
tftp            69/udp
```

Now create a file which starts the BOOTP daemon, e.g. `C:\bin\bootpd.cmd`. A sample could be:

```
/* BOOTP daemon start script */
'@echo off'
say "Starting BOOTPD..."
'bootpd -i 193.141.47.204/255.255.255.240 -d -d -d -d -t 0 -s'
exit 0
```

Note - For further information on the options of the BOOTP and TFTP daemons see the enclosed chapter of the BOOTP and TFTP daemons.

Then create a file to start the TFTP daemon, e.g. `C:\bin\tftpd.cmd`:

```
/* TFTP daemon start script */
'@echo off'
say "Starting TFTPD..."
'tftpd -d /tftpboot -h -r -s 1408 59 -v 2'
exit 0
```

Now create a directory `C:\tftpboot` in which to store all RAM disk images. The directory and all RAM disk images should have read permission for all users.

The format of the `C:\tcpip\etc\bootptab` file is identical to the CMU BOOTP daemon. A sample entry would be:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# sample entry
diskless:\
    :tc=ether:ha=0000c00756bf:ip=193.141.47.198:\
    :hd=/tftpboot:bf=ramd.X:
```

After this start the two CMD-files to load the required daemons.

TFTP security options

Some systems have security options for the TFTP daemon. Check with your manual on how to activate them. Also verify if the BOOTP daemon has the same options.

For example, some operating systems set options like the following in `/etc/inetd.conf`:

```
tftp    dgram   udp   wait    root    /etc/tftpd   tftpd -s /tftpboot
```

This restricts access of TFTP transfers to the directory `/tftpboot`.

Some implementations are known not to work in conjunction with the BOOTP daemon. For example if the user wants to transfer the file `/tftpboot/ramd.X`, the TFTP daemon looks for it in `/tftpboot/tftpboot/ramd.X` instead of `/tftpboot/ramd.X`.

This can be solved by linking the file `/tftpboot/ramd.X` into the directory `/tftpboot/tftpboot/ramd.X`:

```
# mkdir /tftpboot
# chmod a+rx /tftpboot
# cd /tftpboot
# mkdir tftpboot
# chmod a+rx tftpboot
# cp /tmp/ramd.X .
# chmod a+r ramd.X
# ln ramd.X tftpboot
# _
```

Check also if the TFTP daemon is able to use a file `/etc/tftphosts` or `/etc/hosts.tftp` which lists all hosts which are allowed to access TFTP on your system.

Some systems need to create a user tftp in order to enable TFTP transfers.

The BPSHELL Program

The BPSHELL program is an MS-DOS based program that can be used to:

- Create, restore and maintain RAM disk image files.
- Create and maintain BOOTP configuration files.
- Create passwords for diskette drive protection.
- Change parameters of the ROM.
- Create and view menus with the BPMENU system.
- Read TCP/IP BOOT-PROM documentation.

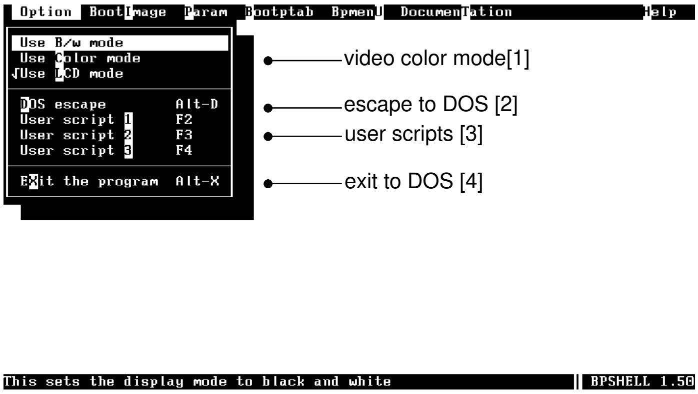
The following chapters describe the BPSHELL menus. The menu can be entered directly by invoking the BPSHELL program with an argument from the command line. The following arguments are accepted:

```
C:\> bpshell -.
  Version: TCP/IP BOOT-PROM 1.50 menu-utility program (bpshell)
Copyright: COPYRIGHT (c) 1989,93 Dirk Köppen EDV-Beratungs-GmbH
Usage    : bpshell [-a|c|e|g|m|n|r|s]
Where    :
        -a   = Encrypt a password
        -c   = Create a RAM disk image
        -e   = Edit a bootptab file
        -g   = Change ROM parameters
        -m   = Modify a bootptab file
        -n   = Read License Agreement
        -r   = Restore a RAM disk image
        -s   = Display a BPMENU script

No argument starts BPSHELL in interactive mode

C:\> _
```

Setting BPSHELL options

Various options can be set for the BPSHELL program. On selecting the

`Option` menu, the following is displayed:



In detail these options are:

[1] The video color mode to be used. On exit this value is stored in the file `BPSHELL.CNF` and read each time the `BPSHELL` program is started. If `BPSHELL.CNF` cannot be read, color mode is used as default.
On color systems, select the color mode. The b/w mode uses shades of gray instead of colors. The LCD mode uses white on black and reverse characters.

[2] The DOS escape temporarily leaves `BPSHELL` for an MS-DOS command interpreter. Enter `exit` at the DOS prompt to return to `BPSHELL`.

[3] User scripts are MS-DOS batch files which can be called via a function key. `BPSHELL` searches for these batch files in the same directory where the `BPSHELL` program is placed. User scripts are named `USER1.BAT`, `USER2.BAT` and `USER3.BAT` depending on the function key used.
These scripts can be used to copy and install RAM disk images to and from the server without leaving `BPSHELL`.

[4] Exit `BPSHELL` and return to MS-DOS.

Creating RAM disk images

The `BPSHELL` program can be used to create RAM disk images. A RAM disk image is a copy of a bootable diskette, it contains all the data of the diskette bundled in a file. Another name for *RAM disk image* is a *container file*.

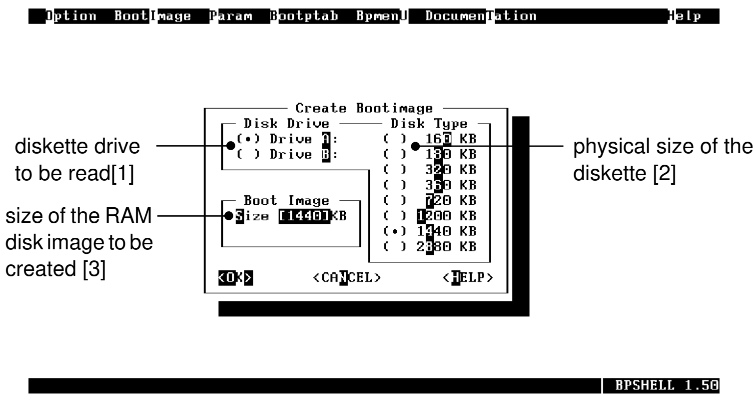
Creating a RAM disk image is done in two steps:

- Create a bootable diskette with the operating system and network software. Samples of how to configure this diskette for various network software is shown in the appendix of this manual.
- Create a RAM disk image file from this diskette. This file is then uploaded and installed on the network server.

The next paragraphs describe how to create a RAM disk image. The procedure can be used to create MS-DOS or other operating system based RAM disk images unless the optimize option is selected. The optimize option only works for MS-DOS systems.

Execute the `BPSHELL` program and select the `Bootimage->Create` window. This window can be entered directly by invoking **`BPSHELL -C`** from the system prompt.

The following window appears:



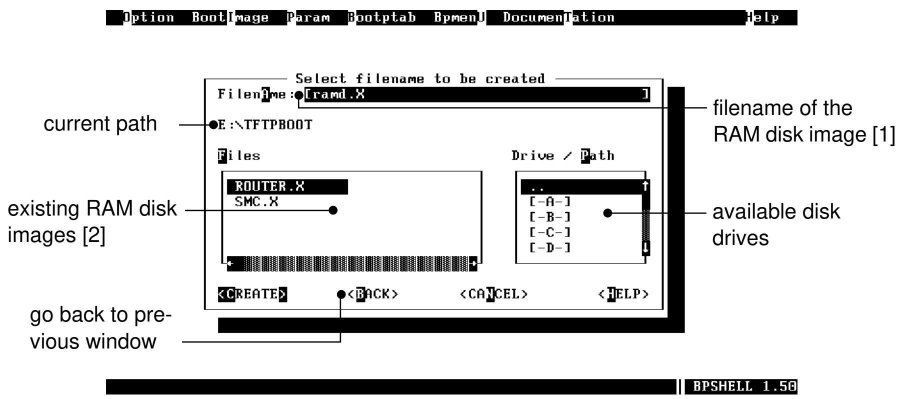
This window permits the selection of the diskette drive and the size to be read. The size of the RAM image can also be set. In detail the options are:

[1] The diskette drive in which the bootable disk is inserted.

[2] The physical size of the diskette. `BPSHELL` checks the bootsector of the diskette to determine its characteristics. This default value may be overwritten by selecting the size which meets the characteristic of your diskette.

[3] In order to decrease the time to download the RAM disk image, make the RAM disk image smaller than the size of the diskette. If a value less than the size of the diskette is entered, `BPSHELL` will only read this number of bytes from the diskette. If a value larger than the size of the diskette is entered, the RAM disk image will be padded with random data.
Take care that all data is moved to the front of the diskette. Otherwise `BPSHELL` may cut off used data. Programs like `DEFRAG` from MS-DOS 6.0 can be used to reorganize a diskette structure.
The default is to create a RAM disk image equal to the maximum size of the diskette.

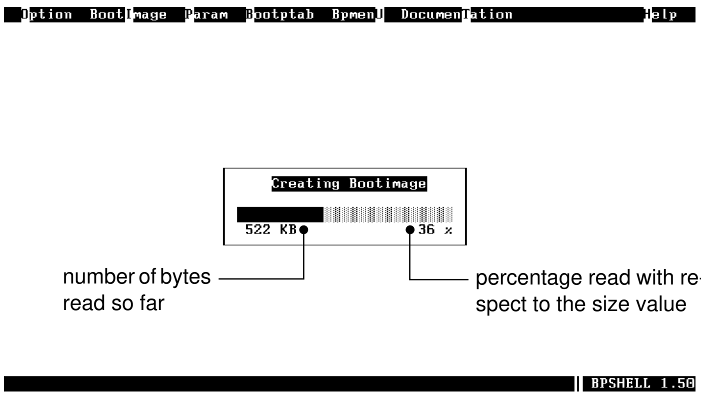
After all selections have been made, press **`<RETURN>`** to accept the window.
The file selection window now comes up:



The name of the RAM disk image to be created is entered in the filename field [1]. To overwrite an existing RAM disk image, select this in the files sub-window [2].

By default `BPSHELL` searches for files with the extension `.X`. This extension directs the TCP/IP BOOT-PROM to load the RAM disk image into Extended Memory. You can disable this by entering something like **`*.IMG`** in the filename field [1].

Now press **`<RETURN>`** to start the RAM disk image procedure:



After the RAM disk image creation is finished you may wish to transfer the RAM disk image file to the server by using `ftp` or a similar file transfer program. You can also create the RAM disk image directly to a mounted network drive.

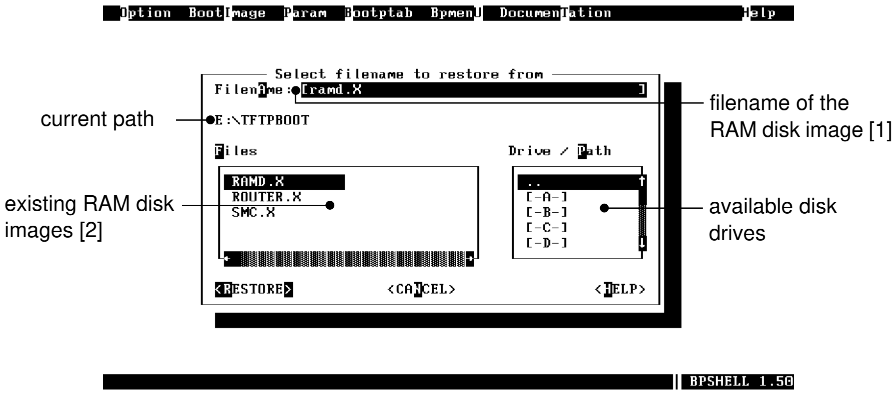
Restoring a RAM disk image

Once a RAM disk image is created it is stored on the server so that the diskless client can access it. If you want to change the RAM disk image either the original diskette is needed or `BPSHELL` can be used to write a RAM disk image onto a diskette.

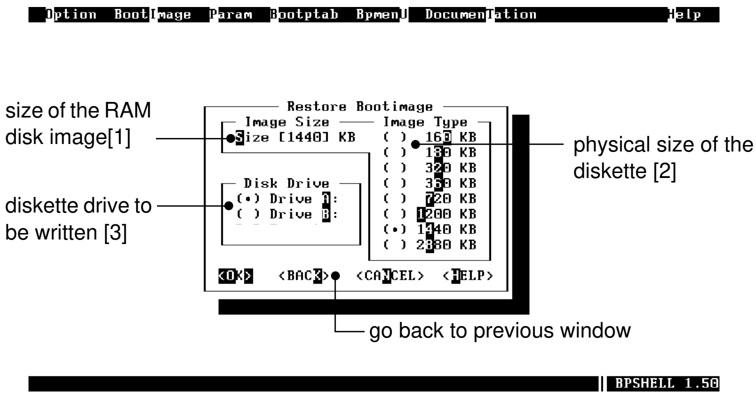
Note - Restoring a RAM disk image to a diskette will overwrite all of the original contents of the diskette.

To restore a RAM disk image invoke the `BPSHELL` program and select the `Bootimage->Restore` window. This window can be entered directly by typing **`BPSHELL -R`** from the system prompt.

The following window will appear:



Select the RAM disk image you want to create and press **<RETURN>** to accept all choices. The diskette characteristics window will now appear:



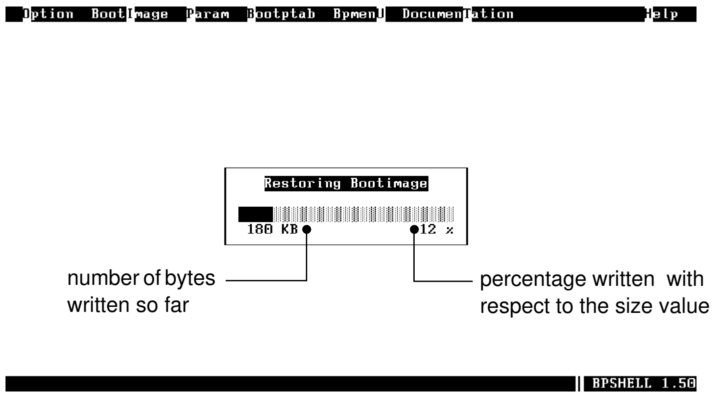
The diskette characteristics window allows you to overwrite the default parameters `BPSHELL` determined while scanning the diskette drive. The parameters are:

[1] The amount of data in K bytes to be restored from the RAM disk image file to the diskette. If the value in the size field is less than the size of the RAM disk image file, the remaining data of the RAM disk image file is ignored.
If the value in the size field is larger than the size of the RAM disk image file, the diskette will be padded with random data.

[2] The physical size of the diskette. `BPSHELL` checks the diskette in order to determine its size. You might want to overwrite this value on unformatted diskettes.

[3] The diskette drive in which the diskette is inserted.

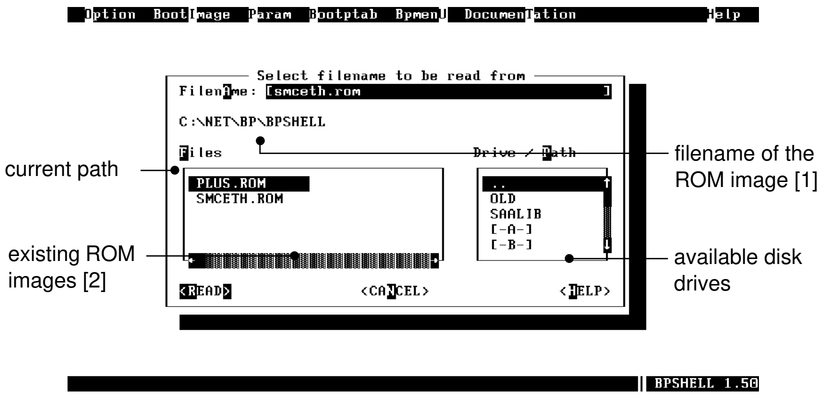
Now press **<RETURN>** to start the restore procedure.



Changing ROM parameters:

The parameters of the ROM can be changed by reading the contents of the ROM into a file. This file is then modified by the `BPSHELL` program and written back into a ROM.

The file selection window can be entered via `Param->Change` or directly by typing `BPSHELL -G` on the system prompt:

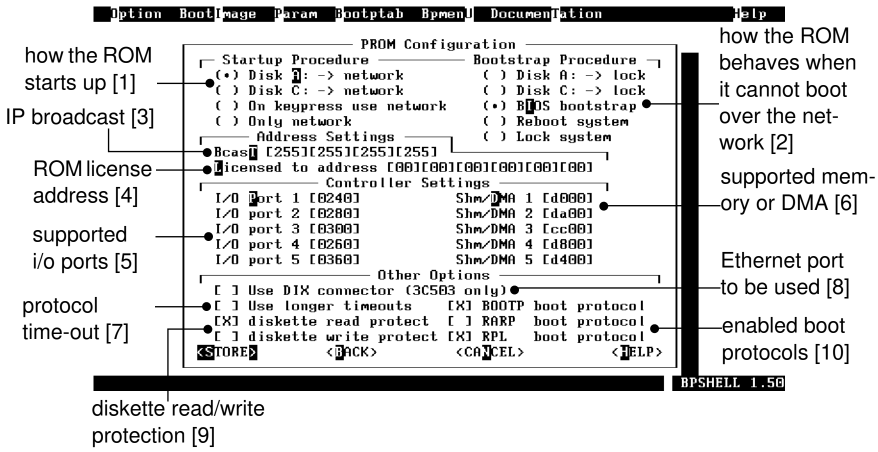


Either enter the name of the ROM image you wish to modify in the filename field [1] or select one from the files sub-window [2]. This must be the name of an existing image.

By default `BPSHELL` searches for files with the extension `.ROM`. You can overwrite this by entering something like **`*.BIN`** in the filename field [1].

Note - The file format of the ROM image is a 1 to 1 binary copy of the contents of the ROM. It does not use any checksumed format like Intel hex.

After pressing **<RETURN>** the parameter window appears:



The options are:

[1] When the TCP/IP BOOT-PROM starts up it checks for bootable drives. The start-up procedure can set where the ROM should look for those disks. The options are:

- Check if drive A: is bootable, if not boot from network.
- Check if drive C: is bootable, if not boot from network.
- Only boot from network if the user presses `<SPACE>`.
- Prevent boot from any drive, allow only network boot.

[2] If the ROM was not able to boot from the network or if the user aborted the network boot by pressing a key, it checks the bootstrap procedure option on how to continue. Several options define the behavior of the ROM:

- Check if drive A: is bootable otherwise lock the system.
- Check if drive C: is bootable otherwise lock the system.
- Execute a regular System BIOS bootstrap.
- Reboot the system. This is retried periodically.
- Lock the system the user must reboot via RESET.

[3] The IP broadcast address defaults to `255.255.255.255` – that is, all bits are set. Older BSD systems use all bits clear for broadcasts which means the broadcast address is `0.0.0.0`. Also you may want to change this setting to a subnet broadcast address.

[4] Some TCP/IP BOOT-PROMs are copy protected. The copy protection sets the network controller's address in the ROM in which case only this controller can be used with the ROM.
This setting cannot be changed.

[5] The TCP/IP BOOT-PROM searches for a network controller at up to 5 i/o port addresses. If the default port addresses do not match your configuration they must be changed.
Unused addresses are set to zero, on auto-detecting network drivers all addresses are set to zero.

[6] If the network driver uses shared memory, all possible shared memory segments can be set here. The ROM scans these segments to determine whether they are allocated to another device, if not the ROM sets the shared memory area to one of these segments.
On network controllers with DMA transfer, this setting is used as the DMA channel.
Unused segments or channels are set to zero. On auto-detecting network drivers all addresses are set to zero.

[7] To prevent time-outs on busy networks or loaded servers the time-out can be increased. If this option is set, the time-out is multiplied by 4 for the TFTP protocols.

[8] This setting is only used on the 3Com 3C503 (Etherlink II) controller. By default the TCP/IP BOOT-PROM uses the BNC connector. If this option is set it uses the DIX connector.

[9] After a successful network boot the diskette drives can be protected. These settings define the type of protection. For read and write protection both options must be set.

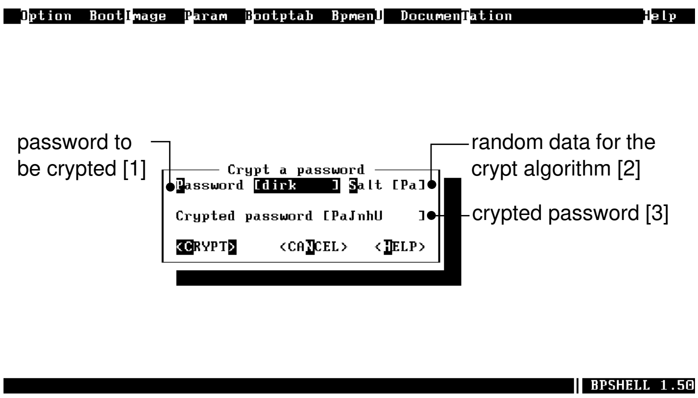
[10] This setting defines the protocol that the TCP/IP BOOT-PROM should use. The RARP and RPL protocols can only be set on the TCP/IP BOOT-PROM *Plus* (obsolete).
If you set multiple protocols, the ROM tries all of them in turn until it gets a reply. It uses the first reply it gets.

After completing all selections, press **`<RETURN>`** to write the changes to the ROM image file.

Encrypting a password

Diskette drives can be read and write protected after a successful network boot. To remove this protection a password must be given to the `BPUTIL` program. These passwords can be generated from the `Param->Crypt` window.

The window can be entered directly by typing `BPSHELL -A` from the system prompt.



To encrypt a password, enter it in the password field [1]. A password can only use the characters `a-z`, `A-Z` and `0-9`.

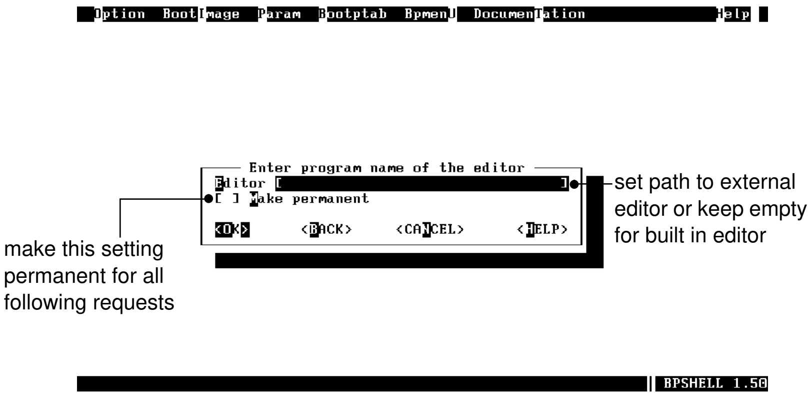
The salt field [2] permits modifications to the password encryption algorithm.
`BPSHELL` will pick out two random characters so that two equal password do not have the same encrypted result.

The string in the encrypted password field [3] can be used as the password argument to `BPUTIL -U U <cpasswd>`.

Selecting the BPSHELL Editor

Files can be edited from `BPSHELL`. The `Bootptab->Edit` menu allows editing `/etc/bootptab` style files, `Bpmenu->Edit` edits `BPMENU` scripts.

If the edit function is started for the first time, the following window appears:



The preferred editor is specified in this window. If the editor field is left empty, `BPSHELL` uses its own built-in editor. Otherwise `BPSHELL` will call the specified external editor every time an edit menu is selected.

Once set, this option is active in memory for all following calls. To make this setting permanent, set the permanent option. This will write the chosen setting into the `BPSHELL.CNF` file which is read on every `subsequent` `BPSHELL` start.

Using the built in-editor

The `BPSHELL` built-in editor has a limited capability to edit ASCII files.

Note - The maximum number of lines the BPSHELL edit can hold is 750.

key	**operation**
`Cursor Left`	Move cursor left one char.
`Ctrl + CurLeft`	Move cursor to previous word.
`Cursor Right`	Move cursor right one char.

key	operation
`Ctrl + CurRight`	Move cursor to next word.
`Cursor Up`	Move cursor up one line.
`Cursor Down`	Move cursor down one line.
`Home`	Move cursor to first character in current line.
`Ctrl + Home`	Move cursor to first line.
`End`	Move cursor to last character in current line.
`Ctrl + End`	Move cursor to last line.
`Ins`	Toggle between insert and delete mode.
`Ctrl + Y`	Delete current line.
`Del`	Delete character under cursor.
`Backspace`	Delete character before cursor.
`F6`	Start/stop selecting text.
`Shift + F6`	Hide text selection.
`Ctrl + K`	Copy selected text to current cursor position.
`Ctrl + L`	Delete selected text.
`Ctrl + V`	Move selected text to current cursor position.
`ESC`	Exit the editor without saving.
`F1`	Show help text.
`F2`	Zoom/unzoom editor window.
`Shift + Tab`	Leave editor and step to next button.

Displaying BPMENU scripts

`BPSHELL` can display `BPMENU` scripts. By selecting the `Bpmenu->Display` windows, selected scripts can be displayed. Typing `BPSHELL -S` from the command enters the `BPMENU` display window directly.

By default all `BPMENU` scripts have the extension `.M`. This must not be changed because the `BPMENU` script interpreter program only accepts this extension.

`BPSHELL` can only interpret the output routines of a `BPMENU` script. It does not allow keyboard input or RAM disk image downloads. This is useful to debug or write `BPMENU` scripts without rebooting the diskless Personal Computer after every change.

After a script is displayed press **`<RETURN>`** to return to `BPSHELL`.

The BPUTIL Program

The `BPUTIL` program is a multi purpose program that works in conjunction with the TCP/IP BOOT-PROM to:

- Restore memory allocated by the TCP/IP BOOT-PROM.
- Patch information given by the BOOTP reply into ASCII and binary files.
- Transfer files from the network server to the client using the TFTP protocol.
- Protect diskette drives after a diskless boot.
- Use the TCP/IP BOOT-PROM with EMM386 and HIMEM.
- Restart the diskless client and other miscellaneous functions.

`BPUTIL` is available as an MS-DOS `COM`-file which can be executed from within batch files. The following sample shows how to restore all memory used by the TCP/IP BOOT-PROM:

```
rem this will free the TCP/IP BOOT-PROM memory space and drive A:...
a:\bin\bputil -r
```

A device driver version is also available. The driver can be loaded from an MS-DOS `CONFIG.SYS` file. After completion the driver releases all its resources so that memory is not permanently allocated by using the driver version of `BPUTIL`. Here is how to use the TCP/IP BOOT-PROM with `EMM386`:

```
dos=high,umb
device=a:\bin\bputil.sys -f
device=a:\himem.sys /testmem:off
device=a:\emm386.exe noems /y=c:\dos\emm386.exe
device=a:\bin\bputil.sys -x
```

BPUTIL command line options

All command line options use the leading character `-`, e.g. `-x`. Options cannot be stacked, i.e. `-x -f`; Multiple calls must be made instead. The following table shows all the `BPUTIL` options in alphabetical order:

command line option	**function**
`-a <fnam>`	Patch BOOTP reply into ASCII file.
`-b <tnum> <fnam>`	Patch BOOTP reply into binary file.
`-c`	Display the TCP/IP BOOT-PROM configuration
`-C`	Set ERRORLEVEL if the PC was not booted over the network.
`-d <dnum>`	Checksum the diskette drive.
`-e`	Reboot the PC via System BIOS call.
`-E`	Reboot the PC via `<Ctrl>-<Alt>-<Del>`.
`-f`	Fix MS-DOS memory allocation for use with `EMM386`.
`-h`	Install the XMS handler entry point.
`-i <fnam>`	Includes a file into the BOOTP reply to extend the BOOTP reply block.
`-i [gip] <sip> <fn>`	Retrieves a file via TFTP and includes it into the BOOTP reply to extend the BOOTP reply block.
`-m`	Display MS-DOS memory control blocks.
`-p <fnam>`	Same as `-a` but uses space to separate IP addresses.
`-r`	Restore TCP/IP BOOT-PROM memory and diskette drive A:.
`-s [tag]`	Display all or a specified tag from the BOOTP reply.
`-S`	As -s but shows only set tags.
`-t <sip> <s> <d>`	TFTP transfer file `s` from network server with the IP address `sip` to file `d`.
`-u r|w|rw|u pwd`	Protect or unprotect diskette drives.
`-v <addr32>`	Relocate the RAM disk to memory address `addr32`.
`-x`	Protect the RAM disk via `HIMEM`.
`-z`	Remove RAM disk but not the TCP/IP BOOT-PROM memory.
`-?`	Shows all BPUTIL options.

Restoring memory used by the TCP/IP BOOT-PROM

The -c option

This gives the location of the BOOTP reply, original System BIOS boot and disk vectors and the RAM size detected by the System BIOS at system start.

```
bootp= 9400:2292 (hex)
bootv= c800:102e (hex)
diskv= c800:10de (hex)
ramds= 00110000 (hex memory)
ramdl= 2791 (512 byte sectors)
ramsz= 640
xms  = 15360
xentr= 0000:0000
xhand= 43262
iop  = 0280 (hex)
shm  = d000 (hex)
dma  = 0000 (hex)
eth  = 0001 (hex)
feat = 821b (hex)
ver  = 1.48
```

The meaning of the fields are:

variable	meaning
`bootp`	Memory location of the BOOTP reply (512 bytes).
`bootv`	Entry point of the original System BIOS boot vector.
`diskv`	Entry point of the original System BIOS disk vector.
`ramds`	Memory location (32 bits address) of the RAM disk data.
`ramdl`	Length of the RAM disk in sectors (512 bytes).
`ramsz`	Original System BIOS Base Memory size.
`xms`	Flag if XMS is used.
`xentr`	Entry point of the XMS handler (`HIMEM`).
`xhand`	XMS handle.
`iop`	I/O port used on the network controller.
`shm`	Shared memory segment used on the network controller.
`dma`	DMA channel used on the network controller.
`eth`	Flag if the 3C503 does use BNC connector.
`feat`	Coded features bits of the TCP/IP BOOT-PROM.
`ver`	The version of the ROM (not the one of BPUTIL).

This option is intended for debugging only. Refer to the TCP/IP BOOT-PROM Technical Reference Manual for further information.

The -C option

The `-C` option checks if the PC was booted over a the network using the TCP/IP BOOT-PROM. If the PC was bootet over the network, the MS-DOS environment variable `ERRORLEVEL` is set to `0`. If the PC was bootet over a disk- or diskette-drive, `ERRORLEVEL` is set to `1`.

The -r option

Using the `-r` option of the `BPUTIL` program removes the RAM disk drive and restores the memory used by the ROM and the RAM disk. The `-r` option also restores the physical diskette drive `A:`.

Note - If `-r` is called, the RAM disk is removed and replaced by the physical diskette drive `A:`. If no diskette is inserted in this drive, MS-DOS will prompt with a read error. To avoid this, only call `-r` from a drive different to `A:`.

If it is not possible to highload programs or the command `MEM /C` no longer displays programs which are loaded into high memory, see the `-f` option for more information.

The -z option

The `-z` option is equivalent to the -r option but does not restore the memory used by the TCP/IP BOOT-PROM. It only restores the memory of the RAM disk and the physical diskette drive.

It is not recommended to use this option because it does not free the Base Memory allocated by the TCP/IP BOOT-PROM.

This option is intended for debugging only.

Patching BOOTP reply information into files

Note - The number of bytes that can be used by tags and pre-defined vendor fields is limited to 64 bytes. This is specified by the RFC1048 BOOTP protocol. `BPUTIL` will warn if the size of the BOOTP reply exceeds the protocol definition. The -i option overcomes this limitation.

The -a option

`BPUTIL -a` patches information given by the BOOTP server into ASCII files. The `BPUTIL` program searches within the given file for a special ID (the string `#@`) and replaces it with a string defined by the tag name following the `@` character.

For example the file

```
# /etc/hosts Internet address database
#
127.0.0.1       localhost local
#@yip########## #@chn##########
```

contains the two tags `#@yip` and `#@chn`. When running `BPUTIL -a` on the file, the two tags will be replaced by strings given by the BOOTP reply. The `#@chn` will be replaced by the hostname of the diskless Personal Computer, the tag `#@yip` will be replaced by the Internet address of the diskless Personal Computer. The length of the field is given by the number of `#` characters and the length of the tag.

If the replacement string is longer than the length of the field, it will cut the string to meet the requested size. If the string is smaller it will be padded with spaces.

After patching the above file with `BPUTIL -a a:\etc\hosts`, the file might look like:

```
# /etc/hosts Internet address database
#
127.0.0.1       localhost local
128.1.1.99      bozo
```

Note - The file name must always contain the drive's letter and the full path (e.g. `a:\etc\hosts`).

The following tags can be used with the `BPUTIL` program:.

tag	RFC951 BOOTP reply
`#@bfn#`	Bootfilename.
`#@cha#`	Client hardware address.
`#@gip#`	BOOTP gateway IP address. If RFC1048 gateway is set, this field contains the first gateway IP address (`gw0`).
`#@shn#`	Server hostname.
`#@sip#`	Server IP address.

tag	RFC951 BOOTP reply
#@yip#	Your (client) IP address.

tag	RFC1048 BOOTP vendor field
#@is0#	First impress server IP address.
#@isf#	All impress server IPs, separated by spaces.
#@ds0#	First domain name server IP address.
#@dsf#	All domain name server IPs, separated by spaces.
#@gw0#	First gateway IP address.
#@gwf#	All gateway IPs, separated by spaces.
#@ls0#	First log server IP address.
#@lsf#	All log server IPs, separated by spaces.
#@lp0#	First line printer server IP address.
#@lpf#	All line printer server IPs, separated by spaces.
#@mdf#	Merit dump file.
#@ns0#	First name server IP address.
#@nsf#	All name server IPs, separated by spaces.
#@qs0#	First quote/cookie server IP address.
#@qsf#	All quote/cookie server IPs, separated by spaces.
#@rs0#	First RLP server IP address.
#@rsf#	All RLP server IPs, separated by spaces.
#@smf#	Subnet mask.
#@tof#	Time offset field.
#@ts0#	Initial server IP address.
#@tsf#	Permanent server IPs, separated by spaces.
#@txxx#	User defined tag number `xxx.`
#@Txxx#	User defined tag number `xxx`, `/` is converted to `\`.

tag	RFC1084 BOOTP vendor field
#@chn#	Client host name.

Tag	RFC1395 BOOTP vendor field
#@cdn#	Client domain name.
#@rfn#	Root pathname.
#@swp#	Swap server IP address.

tag	RFC1497 BOOTP vendor field
#@efn#	Extensions path.

tag	misc. parameters
#@iop#	I/O port used by the PROM (hex).
#@dma#	DMA port used by the PROM (hex).
#@eth#	PROM used BNC (1) or AUI (0) port on 3C503.
#@shm#	Shared memory address used by the PROM (hex).
#@typ#	Type of controller (hex number)[a].

a. See Table , “Supported network controllers,” on page 143 for a listing of all driver numbers.

Note - Not all of the above tags may be defined on your system. The number of available tags depends on the features of the BOOTP daemon used.

The alignment of a string replacing a tag can be specified. Fields like `#@t123#######` are replaced with a left aligned string. A - sign after the tag specifies left and a + sign right alignment.
If no sign is given, left alignment is default. This option can be used to append filename extensions to tags:

```
#@t128##################
#@t128-#################
#@t128+#################
#@t130##.COM
#@t130-#.COM
#@t130+#.COM
```

This will be changed to:

```
/usr5/bp/dos
/usr5/bp/dos
            /usr5/bp/dos
HELLO   .COM
HELLO   .COM
   HELLO.COM
```

The -b option

The `-b` option can be used to patch binary information into a file. The usage of the binary patch option is `-b tagnum filename` where `filename` is the name of the binary file (program, driver...) to be patched. `Tagnum` specifies the tag which contains the information that is to be patched into the file.

The tag is interpreted in a special way. The first three bytes contain in network order (most-significant byte first) the file-offset where the data is to be placed. The following bytes contain the data itself. Note that the file-offset begins with 0. That is 0 is the file-offset of the first byte in a file.

Example: The serial number of the PC-NFS `PCNFS.DRV` driver has to be changed before it is loaded. First find if there are differences between the two `PCNFS.DRV` files using the MS-DOS `FC` command

```
C:\> fc /b pcnfs.1 pcnfs.2
Comparing files PCNFS.1 and PCNFS.2
00000031: 31 32
00000032: 32 23
00000033: 33 36
00000034: 34 33
00000035: 35 33
00000036: 36 34
00000037: 37 37
00000038: 38 36
C:\> _
```

Note - The BPUTIL program does not create or remove serial number protection from programs. It only patches information into files. This information can be a serial number which already exists.

Assume that the serial number starts at file-offset `49` (hex `0x31`) and the serial number is ASCII `12345678`. First, all numbers must be converted to hexadecimal notation and then added to the `/etc/bootptab` file:

```
fileoffset 49 = 000031 (hex)
ASCII 12345678 = 31 32 33 34 35 36 37 38 (hex)
```

thus the entry in the `/etc/booptab` file would be:

```
diskless:\
    :tc=pcnfs:ha=0000c0e23324:ip=193.141.47.198:\
    :bf=pcnfs.X:T132=0000313132333435363738:
```

The `BPUTIL` program must be called in `CONFIG.SYS` in order to patch the serial number before the `PCNFS.DRV` is loaded:

```
device=a:\bin\bputil.sys -b 132 a:\nfs\pcnfs.sys
device=a:\nfs\pcnfs.sys
```

The -i option

The `-i` option includes a file into the BOOTP reply in order to overcome the limited size of the BOOTP vendor field. The format of the included file is the same as the BOOTP vendor field defined in RFC 1497. This option may be used to include a file given by the 'Extensions Path' tag.

BPUTIL option `-t` can be used to transfer a file given by a user tag or by the `efn` tag from BOOTP. Tags that are already defined in the RFC 951 vendor unique field are overwritten by tags in the file, tags that are not defined are added.

The `-i` option can be used in conjunction with the `-a`, `-b`, `-s` and `-S` options. It must me the first option if multiple options are used.

Example:

```
bputil -t 193.141.47.193 /tftpboot/bootp.add bootp.add
bputil -i bootp.add -a a:\startup.bat
```

The -s option

The reply package given by the BOOTP daemon can be displayed by using the `-s` option of the `BPUTIL` program. This option displays all the fields available to the `BPUTIL` program and additionally defined tag fields.
A sample output follows:

```
A:\> bputil -s
set bfn=/usr/bp/img/pcnfs.X
set cdn=
set cha=00.00.c0.3e.3c.40
set chn=opus
set dma=0000
set ds0=
set dsf=
set eth=0001
set gf0=
set gip=193.141.47.193
set gw0=
set gwf=
set iop=0280
set lp0=
set lpf=
set ns0=
set nsf=
set rfn=
set shm=d000
set shn=
set sip=193.141.47.193
set smf=255.255.255.0
set swp=
set tof=
set ts0=
```

```
set tsf=
set typ=0009
set yip=193.141.47.198
set t128=dksoft
set t130=/usr/bp/pcnfs
set t132=[11]0000313233343536373839
set dmp1=638253630104ffffff000304 ... c046f7075738006646b736f6674820d
set dmp2=2f7573722f62702f70636e66 ... 3132233343536373839ff0000000000
A:\> _
```

You can also specify only the variables you want to see:

```
A:\> bputil -s yip sip
set yip=193.141.47.198
set sip=193.141.47.193
A:\> _
```

The format is based on the MS-DOS command interpreter so that environment variables can be easily set:.

```
A:\> bputil -s yip sip > tmp.bat
A:\> call tmp.bat
A:\> _
```

The −s option also displays binary tags. If a tag contains binary characters (less than ASCII 32 or greater than ASCII 127) the tag is displayed as follows:

```
t132= [5] f0023489f1
```

Where [5] indicates that the length of the tag is 5 bytes. The bytes themselves follow in hexadecimal notation.

The last two fields dmp1 and dmp2 of the BPUTIL output show a hexdump of the BOOTP reply vendor field.

The -S option

The −S option lists all tags as the −s option does but only those which contai n data are shown.

The -p option

The −p option is equivalent to the −a option but uses spaces instead of dots to separate IP and hardware addresses e.g. −a produces 193.141.47.193 whereas −p produces 193 141 47 193.

This is useful for Digital's PathWorks.

Transferring files from a TFTP server to a diskless client

The -t and -T options

This option causes a file to be downloaded from a TFTP server. It uses the PROM's built-in TFTP feature to download the file.

Using the -t option the transfer can be aborted by pressing a key. The -T option ignores keyboard input and can therefore not be aborted.

This option should not be used after the network controller driver has been loaded because the `-t` and `-T` options initialize the network controller.

Three arguments are given to the `-t` option. These are:

- Server IP address.
- Server filename.
- Local filename.

The server IP address must be given in decimal format. All filenames are converted to lowercase since the device driver version of `BPUTIL` cannot handle lower/uppercase arguments due to limitations of MS-DOS.

A sample usage follows:

```
device=a:\bin\bputil.sys -t 193.141.47.193 /etc/passwd pwords
```

Protecting diskette drives after a diskless boot

The -u option

Diskette drives can be protected after a diskless boot. Three protection levels are possible:

- Protect for reading.
- Protect for writing.
- Protect for reading and writing.

The level can be set by the `BPSHELL->PARAM->CHANGE` menu. Additionally `BPUTIL` can remove and add protection levels via the `-u` option. Four arguments can be given to the `-u` option:

option	**description**
`-u r`	Add read protection.
`-u w`	Add write protection.

option	description
`-u rw`	Add read and write protection.
`-u u <cpasswd>`	Remove all protections.
`-u u <tagnum>`	

If you add a protection which is already active, the old protection is kept. Protections are defined simultaneously for diskette drive `A:` and `B:`. You cannot protect the TCP/IP BOOT-PROM's RAM disk or fixed disk drives.

A sample for adding read and write protection to both diskette drives in `CONFIG.SYS` file follows:

```
device=a:\bin\bputil.sys -u rw
```

The remove option needs a password for validation. The password is encrypted and can be created from the `BPSHELL->PARAM->CRYPT` menu. The encryption of the password `dirk` is `Pama5si`, thus the encrypted password of `dirk` can be written directly behind the `-u u` option as follows:

```
device=a:\bin\bputil.sys -u u Pama5si
```

or placed in a BOOTP user defined tag.
In the latter case the tag number must be named, i.e.:

```
device=a:\bin\bputil.sys -u u 199
```

and the `/etc/bootptab` should contain an entry such as:

```
diskless:\
    :tc=pcnfs:ha=0000c0e23324:ip=193.141.47.198:\
    :bf=pcnfs.X:T199="Pama5si":
```

In both cases the `BPUTIL` program prompts the user for a password. If the password entered does not match the encrypted one, the protection is not removed:

```
A:\>bputil -u u Pama5si
Enter password : xxxx
BPUTIL: incorrect password !
A:\> _
```

Using the TCP/IP BOOT-PROM with EMM386 and HIMEM

The -f option

If the Upper Memory can no longer be accesed after restoring the RAM disk and diskvectors with the `-r` option (i.e. `MEM /C` does not show

UMBs), then use the `-f` option.

Execute the `-f` option before loading `HIMEM` and `EMM386`:

```
switches=/w
dos=high,umb
device=a:\bin\bputil.sys -f
device=a:\dos\himem.sys /testmem:off
device=a:\dos\emm386.exe NOEMS /y=c:\dos\emm386.exe
```

The `-f` option installs the TCP/IP BOOT-PROM as a TSR program so that `EMM386` does not overwrite its memory space.

The -x option

To prevent other MS-DOS programs from overwriting the RAM disk image in Extended Memory, the option `-x` has been added to the `BPUTIL` program. An extended Memory Service driver must also be installed.

The `-x` option reserves the space used by the RAM disk image, it must be set directly after the XMS driver. A sample fragment of the `CONFIG.SYS` file could be:

```
device=a:\dos\himem.sys /testmem:off
device=a:\bin\bputil.sys -x
```

The `-r` option of the `BPUTIL` program frees any Extended Memory in use. Using the `-x` option on an XT or non-Extended Memory based system causes BPUTIL to ignore this option.

If you do not use the `-x` option, the RAM disk image could be overwritten by other MS-DOS programs. The ROM warns if it was not able to protect the RAM disk image but will continue operating as if the RAM disk image were protected.

Miscellaneous BPUTIL options

The -e and - E options

The `-e` option reboots a Personal Computer by calling the System BIOS reboot routine. The `-E` option reboots the Personal Computer by inserting the `<Ctrl>-<Alt>-<Del>` sequence into the keyboard buffer.

Both options can be used in an MS-DOS batch file if the installation of the network software failed.

The -d option

To verify the RAM disk image, checksum it via the `-d` option. The `BPUTIL` program then checksums the RAM disk in blocks of 10 cylinders. This option is useful to find out whether the RAM disk has been overwritten by another program or not.

For example:

```
device=a:\bin\bputil.sys -d 0
device=a:\memhack.sys
device=a:\bin\bputil.sys -d 0
```

checksums the RAM disk image and then loads a program which may overwrite the memory. The second call to `BPUTIL` gives the checksum of the RAM disk image after the installation of the program `MEMHACK.SYS`. If the checksums differ, the RAM disk should be moved to another location via the `-v` option. The argument to the `-d` option defines the driver on which a checksum is to be made: thus a `0` checksums drive `A:`, a `1` drive `B:`.

The -h option

If the System BIOS memory move function fails, the TCP/IP BOOT-PROM can be configured to use the `HIMEM` memory copy function.

The `-h` option installs the XMS handler entry point to the TCP/IP BOOT-PROM.

This option is intended for debugging only.

The -m option

The MS-DOS memory control blocks can be monitored by the `-m` option:

```
A:\> bputil -m
id=M, maddr=0x000025b0, psp=0x0008, size=0xd53c, len=0x02b0, owner=
id=M, maddr=0x000050c0, psp=0x0008, size=0xd53c, len=0x0004, owner=
id=M, maddr=0x00005110, psp=0x0512, size=0x9400, len=0x00a4, owner=
id=M, maddr=0x00005b60, psp=0x0000, size=0x097f, len=0x0004, owner=
id=M, maddr=0x00005bb0, psp=0x0512, size=0x9400, len=0x0040, owner=
id=M, maddr=0x00005fc0, psp=0x0000, size=0x097f, len=0x0007, owner=
id=M, maddr=0x00006040, psp=0x0605, size=0x0738, len=0x0133, owner=
id=M, maddr=0x00007380, psp=0x097f, size=0x9400, len=0x0008, owner=
id=M, maddr=0x00007410, psp=0x0742, size=0x097e, len=0x023c, owner=
id=M, maddr=0x000097e0, psp=0x097f, size=0x9400, len=0x1000, owner=
id=M, maddr=0x000197f0, psp=0x0000, size=0x097f, len=0x7a80, owner=
id=Z, maddr=0x00094000, psp=0x9400, size=0xfe94, len=0x0bfe, owner=
id=M, maddr=0x0009fff0, psp=0x0008, size=0xd53c, len=0x353a, owner=
```

```
id=M, maddr=0x000d53a0, psp=0x0008, size=0xd53c, len=0x126b, owner=
id=M, maddr=0x000e7a60, psp=0x0000, size=0x097f, len=0x0007, owner=
id=M, maddr=0x000e7ae0, psp=0xe7af, size=0xe8d3, len=0x0124, owner=
id=Z, maddr=0x000e8d30, psp=0x0000, size=0x097f, len=0x072c, owner=
BIOS RAM size= 640 K byte
XMS available= 13624 K byte, XMS largest= 13624 K byte
A:\> _
```

This option in intended for debugging only.

The -v option

The `-v` option is used to move the RAM disk image within the PC's memory. It takes a 32-bit memory address as argument. The RAM disk image is then moved to this location.

For example:

```
device=a:\bin\bputil.sys -v 200000
```

Moves the RAM disk image from its current location to the hexadecimal memory address `0x200000` (2 Mbyte).

The TCP/IP BOOT-PROM Utility Programs

Using BPFLASH

The `BPFLASH` program loads the TCP/IP BOOT-PROM ROM code into the FLASH memory of an Intel EtherExpress 16 FLASH network controller.

Note - `BPFLASH` overwrites the code in the FLASH, therefore a backup should be made before using the `BPFLASH` program. After backup, the original code may be restored at any time using the `BPFLASH` program.

BPFLASH options

The first argument given to the `BPFLASH` program must always be the i/o port where the EtherExpress 16 FLASH resides. The usage is:

```
bpflash io_address [-r fname] [-w fname] [-f] [-d] [-z]|[-e]
```

Where:

argument	meaning
`io_address`	The i/o port to which the network controller is configured.
`-d`	Copy the FLASH pages in reverse order (from page 31 down to page 0).
`-e`	Erase the FLASH.
`-f`	Do not prompt the user for confirmation.
`-r fname`	Read the FLASH and write its contents into the file `filename`.
`-w fname`	Write the contents of the file `filename` into the FLASH.
`-z`	Erase and write zeros into the FLASH.

Creating a FLASH memory backup

A backup of the FLASH can be created by reading its contents into a file:

```
C:\> bpflash 0x300 -r backup.rom
EEPROM settings:
        word 0 = 0x9e2f
        iobase = 0x0300
        membase= 0xd000
        eaddr  = 00.aa.00.2c.f8.7a
        bprom  = enabled (type= 1)
FLASH contents:
        chksum = 0 (UNIX sum), 19451 (UNIX sum -r)
        romsize= 262144 bytes (512 blocks)

FLASH contents has been written to file backup.rom !
C:\> _
```

Writing program code into FLASH memory

The image of the TCP/IP BOOT-PROM is stored in a file. The contents of the file can be written into the FLASH by:

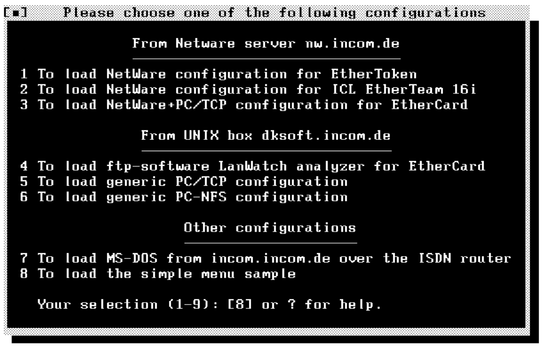
```
C:\> bpflash 0x300 -w exp16f.rom
EEPROM settings:
        word 0 = 0x9e2f
        iobase = 0x0300
        membase= 0xd000
        eaddr  = 00.aa.00.2c.f8.7a.
        bprom  = enabled (type= 1)
Press <RETURN> to overwrite FLASH or Ctrl-C to abort...<RETURN>
FLASH programming:
        zero   = ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
        berase = EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
        write  = WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
        chksum = 0 (UNIX sum), 19451 (UNIX sum -r)
        romsize= 262144 bytes (512 blocks)
FLASH contents:
        chksum = 0 (UNIX sum), 19451 (UNIX sum -r)
        romsize= 262144 bytes (512 blocks)

File exp16f.rom has been successfully written to FLASH !
C:\> _
```

Using BPMENU

`BPMENU` is a menu script interpreter. It can be downloaded and invoked by the TCP/IP BOOT-PROM. The menu prompts the user for a RAM disk image which is then downloaded. Using `BPMENU`, the user can choose between different configurations and servers.

Note - TCP/IP BOOT-PROMs earlier than version 1.47 cannot run `BPMENU`. They must be updated. `BPROML` can be used to update older versions remotely.

Sample BPMENU screen output:



Installing BPMENU

The BPMENU program and a menu script file are held on the server. The BPMENU program must also be specified as the bootfile in the /etc/-bootptab file of the BOOTP daemon. Instead of downloading the RAM disk image, BPMENU is downloaded by the TCP/IP BOOT-PROM.

```
diskless:\
    :tc=pcnfs:ha=00004b0719e1:ip=193.141.47.198:\
    :hd=/tftpboot:bf=bpmenu:
```

After a successful download, control is given to the BPMENU program which now resides in the diskless Personal Computer's memory. A menu script file is now downloaded by the BPMENU program. BPMENU searches for the menu script file by adding the extension .M (dot and capital M) to its name.

For example if BPMENU is downloaded from /tftpboot/bpmenu, then BPMENU uses the filename /tftpboot/bpmenu.M on the same server as its script file.

Note - Optional menu scripts can be accessed by renaming or linking the BPMENU program to a different filename.

After downloading the menu script its contents is interpreted and BPMENU waits for keyboard input.

The BPMENU script language

The BPMENU menu script language has a simple line-by-line command syntax. All commands start with a **.** (dot) in the first column, a 3 character command given by capital letters and a space char if an argument follows. All lines which do not conform to this specification or which have an invalid argument syntax are ignored. This can be used for comments.

```
# clear screen
.CLS

# the selections
.WLN This is a TCP/IP BOOT-PROM menu script, please select:
.WLN
.WLN 1 - will download a RAM disk image into Extended Memory
.KEY 1 192.109.23.193:/tftpboot/image.X
.WLN 2 - will download a RAM disk image into Base Memory
.KEY 2 192.109.23.193:/tftpboot/image

# prompt for input and wait for keypress
.WRT Your choice:
```

Some commands take an optional argument. The argument starts after the space of the command and reaches until the end the of line. Thus appended spaces are also interpreted. To mark this appended spaces the line can be terminated by a \ (backslash) – the backslash is then not interpreted by BP-MENU.

After the menu script has been interpreted, BPMENU waits for input on the keyboard. The action to be taken on a keypress is specified in the menu script.

BPMENU script command

attribute_val
A decimal value (0–255) which defines the colour attributes as listed in the enclosed table.

filename
A string which is interpreted until end-of-line giving the path and name of a file.

ip_addr
An Internet address, each decimal byte is separated by a dot, e.g. 192.109.23.195.

key
One character out of [a-zA-Z0-9?]. Note that no national keyboard mapping is done.

text
A string which is interpreted until end-of-line. A backslash (\) can be used to mark the end of the string if the string ends with space. The terminating backslash will not be printed.

timeout_val
A decimal value which specifies the timeout in seconds.

x, y
A decimal value describing the screen position. x sets the column, y the row.

The commands are:

command	**function**
`.ATT [attribute_val]`	Sets the current character attribute to the value of the optional argument `attribute_val`. If `attribute_val` is not given, the value is toggled back to that one used before the last change.
`.CLS [attribute_val]`	Fills the screen with the character attribute `attribute_val` and sets the cursor to position 0, 0. If no argument is given, the character attribute defaults to white characters on black background (7).
`.DEF key [timeout_val]`	Sets a time-out key. If no key is pressed within `timeout_val` seconds the specified key is used as the users input. If no `timeout_val` is given, 15 seconds are used by default.
`.KEY key [[gip_addr/] ip_addr]:filename`	Defines which file is to be downloaded when the key given by the argument `key` is pressed. If `filename` is tagged by `.M`, this file is downloaded and used as a new menu script. If the `filename` is tagged by `.X`, the file is loaded into Extended Memory. If no `ip_addr` is given or the IP address is `0.0.0.0`, BPMENU uses the former IP address, that is the address from which the last menu script was downloaded. In any case the `:` separator is mandatory. For booting over a router or gateway, its address can be specified by `gip_addr`.
`.POS [x] [y]`	Positions the cursor to position x, y. The upper left corner is 0, 0. If argument y is missing, the cursor is positioned to column x within the current row. If the argument y is also missing, the cursor is placed to the upper left corner.

command	function
.PWD key cpasswd	Sets a password to key. If the user presses key, a password must be given in order to load the configuration. The crypted password cpasswd can be generated using the BPSHELL program.
.WLN [text]	Writes the string given by argument text to the current cursor position using the current character attributes and moves the cursor to the next line. If no argument text is given, the cursor is just moved to the next line.
.WRT text	As .WLN but does not move the cursor to the next line after writing the string given by argument text to the screen.

Note - The maximum size of a menu script is 8192 bytes. The maximum number of key commands allowed is 16.
If the user presses <RETURN>, the default key is used.

Monochrome adapter character attributes

bit coding								function	values
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**		
X								blink	0= no blink
									1= blink
	X	X	X					background	000= black
									111= white
				X				intensity	0= normal
									1= high
					X	X	X	foreground	000= black
									001=underline
									111= white

Example white on black: 00000111 (bin) = 7 (dec)

CGA/EGA/VGA adapter character attributes

7	6	5	4	3	2	1	0	function	values	
X								blink	0=	no blink
									1=	blink
	X	X	X					background	000=	black
									001=	blue
									010=	green
									011=	cyan
									100=	red
									101=	magenta
									110=	brown
									111=	white
				X				intensity	0=	normal
									1=	high
					X	X	X	foreground	000=	black
									001=	blue
									010=	green
									011=	cyan
									100=	red
									101=	magenta
									110=	brown
									111=	white

Example: white on blue is `00011111` (bin) = `31` (dec)

Using BPROML

The `BPROML` program is padded to the code-image of a ROM and can be used to download that ROM code to the diskless Personal Computer. The TCP/IP BOOT-PROM then installs the ROM image in the PC's memory and calls the program code.

`BPROML` is used to update a TCP/IP BOOT-PROM remotely without replacing the ROM device in the PC. Using this technique, all installations within a network can share the same version of the TCP/IP BOOT-PROM and the TCP/IP BOOT-PROM *Utility Programs.*

Note - Only TCP/IP BOOT-PROMs starting at version 1.20 can be updated remotely using BPROML. On version 1.0x and 1.1x the ROM device must be replaced.

Installing BPROML

The BPROML program code must always be copied in front of the ROM code-image. On MS-DOS, copy the BPROML code and ROM code-image in the required order with:

```
C:\> copy /b bproml+wd8isa.rom bproml.U
C:\> _
```

On UNIX systems use:

```
# cat bproml wd8isa.rom > bproml.U
# _
```

After this the resulting image bproml.U must be transferred to the network server.

The file bproml.U is the replacement of the RAM disk image. It must be installed in the /etc/bootptab file. After the bproml.U image has been downloaded by the TCP/IP BOOT-PROM, the bproml.U image is installed in the Personal Computer's memory and executed.

The ROM code-image which is included in bproml.U then removes the .U extension of the bootfilename (e.g. /tftpboot/bproml.U results in /tftpboot/bproml) and uses this filename as the RAM disk image to be downloaded.

Note - Optional menu scripts can be accessed by renaming or linking the BPMENU program to a different filename.

The name of the RAM disk image is always obtained by removing the .U extension of the bootfilename. If multiple extensions are used on the filename e.g. /tftpboot/ramd.XU, only the extension U is removed. The extension U must always be the last character.

Examples of filename replacement:

bootfilename	RAM disk image
/tftpboot/bproml.U	/tftpboot/bproml
/tftpboot/sales.U	/tftpboot/sales
/tftpboot/wd8isa.XU	/tftpboot/wd8isa.X

bootfilename	RAM disk image
/tftpboot/wd8isa.PXU	/tftpboot/wd8isa.PX

Sample BPROML installation

Suppose an existing version 1.45 installation using the WD8ISA ROM is to be updated to the new 1.50 SMCETH ROM code. Assume also that Extended Memory is to be used to upload the RAM disk image.

The old entry in `/etc/bootptab` looks like:

```
diskless:\
    :tc=pctcp:ha=0000c0e91620:\
    :ip=193.141.47.198:\
    :hd=/tftpboot:bf=pctcp.X:\
    :T130="/usr/bp/pctcp":
```

First create the update image containing the new SMCETH ROM image:

```
$ cat bproml smceth.rom > pctcp.U
$ cp pctcp.U /tftpboot/pctcp.XU
$ _
```

Then modify the `/etc/bootptab` entry to:

```
diskless:\
    :tc=pctcp:ha=0000c0e91620:\
    :ip=193.141.47.198:\
    :hd=/tftpboot:bf=pctcp.XU:\
    :T130="/usr/bp/pctcp":
```

The version 1.45 ROM will then download the file `/tftpboot/pctcp.XU` – the remote loadable ROM-image – and execute it. This code will replace the 1.45 ROM and remove the `U` extension from the filename. The resulting file `/tftpboot/pctcp.X` will be downloaded and used as the RAM disk image.

Note - `BPROML` can also be used to call `BPMENU` scripts. TCP/IP BOOT-PROMs earlier than version 1.47 are not able to run `BPMENU`. They can be updated by `BPROML` for use with `BPMENU`.

Using third-party ROM images

Third-party ROM images can also be downloaded and executed. The `BPMENU` program allows the user to choose between several RAM disk images – these can also be `BPROML` programs.

Note - Successful downloading of non-TCP/IP BOOT-PROM based ROM images like NetWare or MOP is not guaranteed. This only works if the ROM image code can be relocated to Base Memory.

A sample LAN WorkPlace installation

This chapter shows a sample installation using the LAN WorkPlace TCP/IP and NFS client network operating system. It is based on the following configuration:

- LAN WorkPlace version 4.1
- NFS client for LAN WorkPlace (B&W NFS) version 2.30
- DR DOS 6.0
- SMC EtherCard Elite with ODI driver
- A UNIX based network server

All the files shown in the following sample are also stored on the TCP/IP BOOT-PROM *Utility Diskette*.

Creating a bootable diskette

At first a bootable diskette must be created which holds the DR DOS operating system, major LAN WorkPlace and NFS client programs to allow a network server logon and programs which are started from `CONFIG.SYS`. All other programs are loaded from the network drive after a successful server logon.

Create a bootable DR DOS diskette:

```
C:\> format a: /s
```

After this copy the DR DOS, LAN WorkPlace and NFS client programs to your diskette. The files `BPUTIL.COM` and `BPUTIL.SYS` are on the TC-P/IP BOOT-PROM *Utility Diskette*. The following table shows which files are needed on the diskette:

```
C:\> chkdsk a:\ /v
Volume LWPNFS created 09-03-1993 1:55p
Directory A:\
```

```
A:\IBMBIO.COM
A:\IBMDOS.COM
A:\COMMAND.COM
Directory A:\DRDOS
A:\DRDOS\KEYB.COM
A:\DRDOS\EMM386.SYS
A:\DRDOS\SUBST.EXE
Directory A:\BIN
A:\BIN\BPUTIL.COM
A:\BIN\BPUTIL.SYS
Directory A:\ODI
A:\ODI\SMC8000.COM
A:\ODI\LSL.COM
A:\ODI\NET.CFG
Directory A:\NET
Directory A:\NET\BIN
A:\NET\BIN\TCPIP.EXE
A:\NET\BIN\LWPRPC.COM
A:\NET\BIN\LWPNFS.COM
A:\NET\BIN\ETHDEV.LWP
A:\NET\BIN\NET.EXE
Directory A:\NET\TCP
A:\NET\TCP\NETWORKS
A:\NET\TCP\PROTOCOL
A:\NET\TCP\SERVICES
A:\NET\TCP\HOSTS
A:\CONFIG.SYS
A:\AUTOEXEC.BAT
Directory A:\TMP
C:\> _
```

Edit the file `CONFIG.SYS` to contain:

```
hidos=on

device=a:\bin\bputil.sys -f
device=a:\drdos\emm386.sys /frame=none /exclude=d000-d3ff /bdos=ffff

buffers=5
files=40
shell=a:\command.com /p /e:1024
hidevice=a:\net\bin\ethdev.lwp GMT 193.1.1.4 16000
lastdrive=V
```

Edit the file `AUTOEXEC.BAT` to include the following lines:

```
@echo off

rem setup the search path for the host directories...
set PATH=a:\;a:\drdos;a:\net\bin;c:\net\bin
```

```
set COMSPEC=a:\command.com

rem make LWP believe it's started from drive C:...
subst c: a:\
c:
cd \

rem usual AUTOEXEC.BAT stuff starts here...
set TZ=MEZ0
keyb gr+

rem set additional LAN WorkPlace parameters...
set NAME=dirk

rem start ODI workstation...
cd \odi
lsl
hiload smc8000
cd \

rem start LAN WorkPlace...
hiload tcpip.exe

rem start NFS client for LAN WorkPlace...
hiload lwprpc
hiload lwpnfs icl /r:8192 /w:8192

rem mount C: drive...
a:
cd \
subst c: /d
:NetworkLoop
net register
net link c: \\193.1.1.4\/demo

rem test for network drive...
if exist c:\autoexec.bat goto NetworkOk
echo Failed to mount network drive !
goto NetworkLoop

rem switch over to network drive...
:NetworkOk
c:
cd \
autoexec.bat
```

Note - This example uses the server and client IP addresses directly in the files. For more flexibility you may want to use the BPUTIL program to patch BOOTP information into configuration files.

The file ODI\NET.CFG loads the network driver and could look like:

```
Link Support
      Max Stacks 8
      Buffers 8 2048
      MemPool 16384
      Max Boards 8

Link Driver SMC8000
      Frame Ethernet_II
      Frame Ethernet_802.2
      Frame Ethernet_802.3

Protocol TCPIP
      PATH HOSTS c:\NET\TCP\HOSTS
      PATH SCRIPT c:\NET\SCRIPT
      PATH PROFILE c:\NET\PROFILE
      PATH LWP_CFG c:\NET\HSTACC
      PATH TCP_CFG c:\NET\TCP
      ip_router 193.1.1.4
      ip_netmask 255.255.255.0
      ip_address 193.1.1.8
```

The file NET\TCP\HOSTS holds the IP addresses of your site:

```
127.0.0.1 localhost
193.1.1.1 nwserver
193.1.1.4 osney peacock
193.1.1.8 icl osg193
```

Creating and installing a RAM disk image

The RAM disk image is a copy of the diskette just created.

Use the program BPSHELL to create that copy on MS-DOS systems. See “Creating RAM disk images” on page 55 for how to create RAM disk images. After a RAM disk image has been created, transfer it to the network server with ftp or copy the file to a mounted network drive. In the following procedure we assume that the RAM disk image is placed in /tftpboot/ramd.X[1] on the network server.

1. /tftpboot is a common place for diskless client files on UNIX systems. The extension .X indicates that the TCP/IP BOOT-PROM should load the RAM disk image into Extended Memory.

If you are working on a UNIX system the following command can be used:

```
# dd if=/dev/rfd0b of=/tftpboot/ramd.X bs=16k
# _
```

This assumes that /dev/rfd0b is your floppy drive, the argument bs=16k is only needed to speed up the floppy disk transfer and can be omitted.

Configuring the host files

The RAM disk image must be readable for all users. Check that the directory and the file have read permission. If not, you can set them by:

```
# chmod 755 /tftpboot
# ls -ld /tftpboot
drwxr-xr-x   2 root     other       32 Aug 15 13:32 /tftpboot
# chmod 744 /tftpboot/ramd.X
# ls -l /tftpboot/ramd.X
-rwxr--r--   1 root     other  1428480 Aug 15 13:33 /tftpboot/ramd.X
# _
```

After this, a user must be created on the network server and all relevant files copied into the directory of that user. These files include the complete DR DOS, LAN WorkPlace and NFS client distribution as well as files needed for the user's application(s). Copy COMMAND.COM and AUTOEXEC.BAT to the home directory of the user. A possible file structure could look like:

```
# ls -l /usr/lwpnfs
total 126
-rw-r--r--   1 lwpnfs    msdos       1112 Aug 15 13:41 autoexec.bat
-rw-r--r--   1 lwpnfs    msdos      52925 Aug 15 13:41 command.com
drwxr-xr-x   2 lwpnfs    msdos       2608 Aug 15 13:42 drdos
drwxr-xr-x   5 lwpnfs    msdos       1344 Aug 15 13:42 net
drwxr-xr-x   5 lwpnfs    msdos       1024 Aug 15 13:42 nfsclnt
drwxr-xr-x   5 lwpnfs    msdos         96 Aug 15 13:42 odi
drwxr-xr-x   2 lwpnfs    msdos         64 Aug 15 13:43 tmp
drwxr-xr-x   4 lwpnfs    msdos       2000 Aug 15 13:43 windows
# _
```

Edit the file AUTOEXEC.BAT to add some site specific commands:

```
@echo off

rem setup the search path for the host directories...
set PATH=c:\;c:\drdos;c:\net\bin;c:\nfsclnt
set COMSPEC=c:\command.com

rem this will free the TCP/IP BOOT-PROM memory space and drive A:...
a:\bin\bputil -r

rem usual AUTOEXEC.BAT stuff starts here...
prompt $p$g
```

```
set PATH=%PATH%;c:\windows
set TMP=c:\tmp
set TEMP=c:\tmp

rem my private environment...
hiload mouse
ver
```

Note - If DR DOS files are edited on a UNIX system, check that all lines are terminated by `CR/NL`. The program `UNIX2DOS` converts files to be DOS conform.

Once again check if the user can access all files in his home directory. A simple way to set the owner of all files would be:

```
# find /usr/lwpnfs -print | xargs chown lwpnfs.msdos
# chown lwpnfs.msdos /usr/lwpnfs
# _
```

Configuring the BOOTP daemon

Now the BOOTP daemon configuration must be set up. Check with the chapter “Installing BOOTP and TFTP” on page 37 to see if there are any specific requirements for your operating system. Also check that the TFTP daemon is enabled.

A possible `/etc/bootptab` entry would be:

```
# template for vanilla Ethernet class C network
ether:\
      :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# template configuration for LAN WorkPlace
lwpnfs:\
      :tc=ether:hd=/tftpboot:

# diskless client
diskless:\
      :tc=lwpnfs:ha=0000c0595f66:ip=193.1.1.8:\
      :bf=ramd.X:
```

Note - Some older and less functional BOOTP daemons have different formats for the `/etc/bootptab` file or a different place and name of their BOOTP daemon configuration file. Check your manual or use the BOOTP daemon on the TCP/IP BOOT-PROM *Utility Disk*.

Testing the configuration

The configuration is now ready for use. If problems are encountered, check with the chapter “Troubleshooting and techniques” on page 137 and also with “Advanced techniques” on page 127 for more tips on how to configure diskless systems using the TCP/IP BOOT-PROM.

A sample NetWare installation

This chapter shows a sample installation using the NetWare network operating system. It is based on the following configuration:

- NetWare shell version 3.31
- DR DOS 6.0
- SMC EtherCard Elite with ODI driver
- A NetWare based file server

All the files shown in the following sample are also stored on the TCP/IP BOOT-PROM *Utility Diskette*.

Creating a bootable diskette

At first a bootable diskette must be created which holds the DR DOS operating system, major NetWare client programs to allow a network server logon and programs which are started from `CONFIG.SYS`. All other programs are loaded from the network drive after a successful server logon.

Create a bootable DR DOS diskette:

```
C:\> format a: /s
```

After this copy the DR DOS and NetWare client programs to your diskette. The files `BPUTIL.COM` and `BPUTIL.SYS` are on the TCP/IP BOOT-PROM *Utility Diskette*. The following table shows which files are needed on the diskette:

```
C:\> chkdsk a:\ /v
Volume NETWARE created 09-03-1993 1:55p
Directory A:\
A:\IBMBIO.COM
A:\IBMDOS.COM
A:\COMMAND.COM
Directory A:\DRDOS
```

```
A:\DRDOS\KEYB.COM
A:\DRDOS\EMM386.SYS
Directory A:\BIN
A:\BIN\BPUTIL.COM
A:\BIN\BPUTIL.SYS
Directory A:\ODI
A:\ODI\SMC8000.COM
A:\ODI\LSL.COM
A:\ODI\NET.CFG
A:\ODI\IPXODI.COM
A:\ODI\NETX.EXE
A:\AUTOEXEC.BAT
A:\CONFIG.SYS
Directory A:\TMP
C:\> _
```

Edit the file `CONFIG.SYS` to contain:

```
hidos=on

device=a:\bin\bputil.sys -f
device=a:\drdos\emm386.sys /frame=none /exclude=d000-d3ff /bdos=ffff

buffers=5
files=40
shell=a:\command.com /p /e:1024
```

Edit the file `AUTOEXEC.BAT` to include the following lines:

```
@echo off

rem setup the search path for the host directories...
set PATH=a:\;a:\drdos;a:\odi
set COMSPEC=a:\command.com

rem usual AUTOEXEC.BAT stuff starts here...
set TZ=MEZ0
keyb gr+

rem start ODI workstation...
cd \odi
lsl
hiload smc8000
hiload ipxodi
cd \

rem mount F: drive...
:NetworkLoop
hiload netx

```

```
rem test for network drive...
if exist f:login.exe goto NetworkOk
echo Failed to mount network drive !
goto NetworkLoop

rem switch over to network drive...
:NetworkOk
f:
login

rem test for home directory...
if exist c:autoexec.bat goto HomeOk
echo Failed to mount home directory !
goto NetworkLoop

rem switch over to HOME directory...
:HomeOk
c:
autoexec.bat
```

The file `ODI\NET.CFG` loads the network driver and could look like:

```
Link Driver SMC8000
      Frame Ethernet_II
      Frame Ethernet_802.2
      Frame Ethernet_802.3
 Protocol IPX E0 Ethernet_802.2
```

Creating and installing a RAM disk image

The RAM disk image is a copy of the diskette just created.

Use the program `BPSHELL` to create that copy. See “Creating RAM disk images” on page 55 for how to create RAM disk images. After a RAM disk image has been created, transfer the file to the network server. In the following procedure we assume that the RAM disk image is placed in `sys:/tftpboot/ramd.X`[1] on the network server.

Note - Take care that all users have permission to read the RAM disk image.

---

1. /tftpboot is a common place for diskless client files on UNIX systems. The extension .X indicates that the TCP/IP BOOT-PROM should load the RAM disk image into Extended Memory.

Configuring the host files

After this, a user must be created on the network server and all relevant files copied into the directory of that user. These files include the complete DR DOS and NetWare client distribution as well as files needed for the user's application(s). Copy `COMMAND.COM` and `AUTOEXEC.BAT` to the home directory of the user. A possible file structure could look like:

```
COMMAND COM      52925 3-10-93  6:00a
AUTOEXEC BAT       392 9-04-93 11:58p
DRDOS            <DIR> 9-05-93 12:15a
WINDOWS          <DIR> 9-05-93 12:00a
TEMP             <DIR> 9-05-93 12:06a
```

The user should have a login script that maps its directory to the drive `C:`. A possible extract out of the login script may be:

```
map root c: = sys:\users\dos
```

Additionally a secondary `AUTOEXEC.BAT` in the users directory should be created:

```
@echo off

rem setup the search path for the host directories...
set PATH=c:\;c:\drdos;f:\public
set COMSPEC=c:\command.com

rem this will free the TCP/IP BOOT-PROM memory space and drive A:...
a:\bin\bputil -r

rem usual AUTOEXEC.BAT stuff starts here...
prompt $p$g
set PATH=%PATH%;c:\windows
set TMP=c:\tmp
set TEMP=c:\tmp

rem my private environment...
hiload mouse
ver
```

Configuring the BOOTP daemon

Now the BOOTP daemon configuration must be set up. Check with the chapter "Installing BOOTP and TFTP" on page 37 to see how to install the BOOTP and TFTP NLMs on NetWare.

A possible `sys:/etc/bootptab` entry would be:

```
# template for vanilla Ethernet class C network
ether:\
      :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# template configuration for NetWare
nware:\
      :tc=ether:hd="SYS:/TFTPBOOT":

# diskless client
diskless:\
      :tc=nware:ha=0000c0595f66:ip=193.1.1.8:\
      :bf=ramd.X:
```

Testing the configuration

The configuration is now ready for use. If problems are encountered, check with the chapter “Troubleshooting and techniques” on page 137 and also with “Advanced techniques” on page 127 for more tips on how to configure diskless systems using the TCP/IP BOOT-PROM.

A sample PC-Interface installation

This chapter shows a sample installation using the PC-Interface network operating system. It is based on the following configuration:

- PC-Interface version 4.1.3
- MS-DOS 6.0
- SMC EtherCard Elite with PC-Interface driver
- A UNIX based network server

All the files shown in the following sample are also stored on the TCP/IP BOOT-PROM *Utility Diskette*.

Creating a bootable diskette

At first a bootable diskette must be created which holds the MS-DOS operating system, major PC-Interface programs to allow a network server logon and programs which are started from `CONFIG.SYS`. All other programs are loaded from the network drive after a successful server logon.

Create a bootable MS-DOS diskette:

```
C:\> format a: /u/s
```

After this copy the MS-DOS and PC-Interface programs to your diskette. The files `BPUTIL.COM` and `BPUTIL.SYS` are on the TCP/IP BOOT-PROM *Utility Diskette*. The following table shows which files are needed on the diskette:

```
C:\> chkdsk a:\ /v
Volume PCI_SAMPL created 05-22-1993 6:58p
Volume Serial Number is 1AF7-3571
Directory A:\
A:\IO.SYS
A:\MSDOS.SYS
A:\COMMAND.COM
```

```
A:\CONFIG.SYS
A:\AUTOEXEC.BAT
Directory A:\PCI
A:\PCI\PCIINST.INI
Directory A:\PCI\BIN
A:\PCI\BIN\BRIDGE.DRV
A:\PCI\BIN\BRIDGE.DAT
A:\PCI\BIN\LOGIN.EXE
A:\PCI\BIN\LOGOUT.EXE
A:\PCI\BIN\PCIINIT.EXE
Directory A:\PCI\DRIVERS
A:\PCI\DRIVERS\WD80X3.DRV
Directory A:\PCI\LIB
A:\PCI\LIB\PC437.LCS
A:\PCI\LIB\PC850.LCS
A:\PCI\LIB\8859.LCS
Directory A:\DOS
A:\DOS\EMM386.EXE
A:\DOS\HIMEM.SYS
A:\DOS\SUBST.EXE
Directory A:\BIN
A:\BIN\BPUTIL.COM
A:\BIN\BPUTIL.SYS
Directory A:\ETC
A:\ETC\HOSTS
C:\> _
```

Edit the file `CONFIG.SYS` to contain:

```
switches=/w
dos=high,umb

rem you may patch your PCI serial number here...
rem device=a:\bin\bputil.sys -b 132 a:\pci\bin\bridge.dat
device=a:\bin\bputil.sys -a a:\pci\pciinst.ini

device=a:\bin\bputil.sys -f
device=a:\dos\himem.sys /testmem:off
device=a:\bin\bputil.sys -x
device=a:\dos\emm386.exe /x=d000-d3ff NOEMS /y=c:\dos\emm386.exe

buffers=5
files=40
shell=a:\command.com /p /e:1024

devicehigh=a:\pci\drivers\wd80x3.drv /i:280 /x:5 /m:d0000
devicehigh=a:\pci\bin\bridge.drv

lastdrive=V
```

Edit the file AUTOEXEC.BAT to include the following lines:

```
@echo off

rem setup the search path for the host directories...
set PATH=a:\;a:\dos;a:\pci\bin
set COMSPEC=a:\command.com

rem make PCI believe it's started from drive C:...
subst c: a:\
c:

rem usual AUTOEXEC.BAT stuff starts here...
set TZ=MEZ0

rem set additional Sun Microsystems PC-NFS parameters...
set HOSTS=c:\etc\
set EMPATH=c:\pci
set LCSPATH=c:\pci\lib
set NLSPATH=c:\pci\%%N\%%L.%%C
set LANG=En

rem patch the BOOTP reply into PCI configuration files...
a:\bin\bputil -a a:\etc\hosts

rem start PCI...
pciinit

:NetworkLoop
login /D:d

rem test for network drive...
if exist d:autoexec.bat goto NetworkOk
echo Failed to mount network drive !
goto NetworkLoop

rem switch over to network drive...
:NetworkOk
d:
autoexec.bat
```

Create the file ETC\HOSTS, all fields starting with #@# will be set later when system is started by the TCP/IP BOOT-PROM:

```
127.0.0.1           localhost local
#@yip#########      #@chn########################
```

Edit the file `PCI\PCIINST.INI` which configures the PC-Interface software:

```
[install]
 product = PC-Interface
 release = 4.1.3
 version = 412
 windir = C:\WINDOWS\
 brgif = pci
 netcard = wd80x3
 netif = pcidrv
 dosver = 600
 cpu = 486
 srcdrv = A:\

[components]
 base = yes
 utilities = yes
 nco = no

[config]
 config = C:\CONFIG.SYS
 autoexec = C:\AUTOEXEC.BAT
 modtype = modactual

[wd80x3]
 ioaddr = 280
 irq = 5
 memaddr5 = d0000
 drvfile = wd80x3.drv
 params = ioaddr irq memaddr5

[network]
 pcname = #@chn##################
 ipaddr = #@yip##################
 subnetbits = 24
 subnetmask = 255.255.255.0
 frame = ethernet_ii

[pciinit]
 pcname = #@chn##################
 ipaddr = #@yip##################
 subnetmask = 255.255.255.0

[nvllwp]
 drvfile = lcclwp.drv
 params =
```

Creating and installing a RAM disk image

The RAM disk image is a copy of the diskette just created.

Use the program `BPSHELL` to create that copy on MS-DOS systems. See “Creating RAM disk images” on page 55 for how to create RAM disk images. After a RAM disk image has been created, transfer it to the network server by copying the file to a mounted network drive. In the following procedure we assume that the RAM disk image is placed in `/tftpboot/ramd.X`[1] on the network server.

If you are working on a UNIX system the following command can be used:

```
# dd if=/dev/rfd0b of=/tftpboot/ramd.X bs=16k
# _
```

This assumes that `/dev/rfd0b` is your floppy drive, the argument `bs=16k` is only needed to speed up the floppy disk transfer and can be omitted.

Configuring the host files

The RAM disk image must be readable for all users. Check that the directory and the file have read permission. If not, you can set them by:

```
# chmod 755 /tftpboot
# ls -ld /tftpboot
drwxr-xr-x   2 root     other       32 Aug 15 13:32 /tftpboot
# chmod 744 /tftpboot/ramd.X
# ls -l /tftpboot/ramd.X
-rwxr--r--   1 root     other  1428480 Aug 15 13:33 /tftpboot/ramd.X
# _
```

1. /tftpboot is a common place for diskless client files on UNIX systems. The extension .X indicates that the TCP/IP BOOT-PROM should load the RAM disk image into Extended Memory.

After this, a user must be created on the network server and all relevant files copied into the directory of that user. These files include the complete MS-DOS and PC-Interface distribution as well as files needed for the user's application(s). Copy `COMMAND.COM` and `AUTOEXEC.BAT` to the home directory of the user. A possible file structure could look like:

```
# ls -l /usr/bp/pci
total 126
-rw-r--r--   1 pci    msdos       1112 Aug 15 13:41 autoexec.bat
-rw-r--r--   1 pci    msdos      52925 Aug 15 13:41 command.com
drwxr-xr-x   2 pci    msdos       2608 Aug 15 13:42 dos
drwxr-xr-x   2 pci    msdos         64 Aug 15 13:43 etc
drwxr-xr-x   5 pci    msdos       1344 Aug 15 13:42 pci
drwxr-xr-x   2 pci    msdos         64 Aug 15 13:43 tmp
drwxr-xr-x   4 pci    msdos       2000 Aug 15 13:43 windows
# _
```

Edit the file `AUTOEXEC.BAT` to add some site specific commands:

```
@echo off

rem setup the search path for the host directories...
a:\dos\subst c: /d
a:\dos\subst c: d:\usr\bp\pci
c:
set PATH=c:\;c:\dos;c:\pci\bin
set COMSPEC=c:\command.com

rem this will free the TCP/IP BOOT-PROM memory space and drive A:...
a:\bin\bputil -r

rem usual AUTOEXEC.BAT stuff starts here...
prompt $p$g
set PATH=%PATH%;c:\windows
set TMP=c:\tmp
set TMPDIR=c:\tmp

rem additional PCI commands...
if exist c:\pci\em.ses del c:\pci\em.ses

rem my private environment...
loadhigh keyb gr
loadhigh mouse
ver
```

Note - If MS-DOS files are edited on a UNIX system, check that all lines are terminated by `CR/NL`. The program `UNIX2DOS` converts files to be DOS conform.

Once again check if the user can access all files in his home directory. A simple way to set the owner of all files would be:

```
# find /usr/bp/pci -print | xargs chown pci.msdos
# chown pci.msdos /usr/bp/pci
# _
```

Configuring the BOOTP daemon

Now the BOOTP daemon configuration must be set up. Check with the chapter “Installing BOOTP and TFTP” on page 37 to see if there are any specific requirements for your operating system. Also check that the TFTP daemon is enabled.

A possible `/etc/bootptab` entry would be:

```
# template for vanilla Ethernet class C network
ether:\
      :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# template configuration for PC-Interface
pci:\
      :tc=ether:hd=/tftpboot:

# diskless client
diskless:\
      :tc=pci:ha=0000c0595f66:ip=193.141.47.198:\
      :bf=ramd.X:
```

Note - Some older and less functional BOOTP daemons have different formats for the `/etc/bootptab` file or a different place and name of their BOOTP daemon configuration file. Check your manual or use the BOOTP daemon on the TCP/IP BOOT-PROM *Utility Disk*.

Testing the configuration

The configuration is now ready for use. If problems are encountered, check with the chapter “Troubleshooting and techniques” on page 137 and also with “Advanced techniques” on page 127 for more tips on how to configure diskless systems using the TCP/IP BOOT-PROM.

A sample PC-NFS installation

This chapter shows a sample installation using the PC-NFS network operating system. It is based on the following configuration:

- PC-NFS version 5.0
- MS-DOS 6.0
- SMC EtherCard Elite with ODI driver
- A UNIX based network server

All the files shown in the following sample are also stored on the TCP/IP BOOT-PROM *Utility Diskette*.

Creating a bootable diskette

At first a bootable diskette must be created which holds the MS-DOS operating system, major PC-NFS programs to allow a network server logon and programs which are started from `CONFIG.SYS`. All other programs are loaded from the network drive after a successful server logon.

Create a bootable MS-DOS diskette:

```
C:\> format a: /u/s
```

After this copy the MS-DOS and PC-NFS programs to your diskette. The files `BPUTIL.COM` and `BPUTIL.SYS` are on the TCP/IP BOOT-PROM *Utility Diskette*. The following table shows which files are needed on the diskette:

```
C:\> chkdsk a:\ /v
Volume PCNFS_SAMPL created 22 May 1993 18.58
Directory A:\
 A:\IO.SYS
 A:\MSDOS.SYS
 A:\COMMAND.COM
 A:\CONFIG.SYS
```

```
 A:\AUTOEXEC.BAT
Directory A:\DOS
 A:\DOS\EMM386.EXE
 A:\DOS\HIMEM.SYS
Directory A:\BIN
 A:\BIN\BPUTIL.COM
 A:\BIN\BPUTIL.SYS
Directory A:\NFS
 A:\NFS\DRIVES.BAT
 A:\NFS\ETHERS
 A:\NFS\HOSTS
 A:\NFS\NET.EXE
 A:\NFS\NETCONF
 A:\NFS\NETWORK.BAT
 A:\NFS\NETWORKS
 A:\NFS\PASSWD
 A:\NFS\PCNFS.MSG
 A:\NFS\PCNFS.SYS
 A:\NFS\PROTOS
 A:\NFS\PRT.COM
 A:\NFS\NFSODI.SYS
 A:\NFS\RPC
 A:\NFS\RTM.EXE
 A:\NFS\SERVICES
 A:\NFS\SOCKDRV.SYS
 A:\NFS\RNMFILE.EXE
 A:\NFS\FTPUSERS
Directory A:\ODI
 A:\ODI\LSL.COM
 A:\ODI\SMC8000.COM
 A:\ODI\NET.CFG
C:\> _
```

Edit the file `CONFIG.SYS` to contain:

```
switches=/w
dos=high,umb

device=a:\bin\bputil.sys -f
device=a:\dos\himem.sys /testmem:off
device=a:\bin\bputil.sys -x
device=a:\dos\emm386.exe /x=d000-d3ff NOEMS /y=c:\dos\emm386.exe

buffers=5
files=40
shell=a:\command.com /p /e:1024
devicehigh=a:\nfs\pcnfs.sys /F40
devicehigh=a:\nfs\sockdrv.sys
devicehigh=a:\nfs\nfsodi.sys
lastdrive=V
```

Edit the file AUTOEXEC.BAT to include the following lines:

```
@echo off

rem setup the search path for the host directories...
set PATH=a:\;a:\dos;a:\nfs
set COMSPEC=a:\command.com

rem usual AUTOEXEC.BAT stuff starts here...
set TZ=MEZ0

rem set additional PC-NFS parameters...
set NFSDRIVE=a
set NFSPATH=a:\nfs

rem patch the BOOTP reply into PC-NFS configuration files...
a:\bin\bputil -a a:\nfs\hosts
a:\bin\bputil -a a:\nfs\drives.bat
a:\bin\bputil -a a:\nfs\network.bat

rem start ODI workstation...
cd \odi
lsl
smc8000
cd \

rem start PC-NFS...
loadhigh prt *
:NetworkLoop
net init

rem test for network drive...
if exist c:\autoexec.bat goto NetworkOk
echo Failed to mount network drive !
goto NetworkLoop

rem switch over to network drive...
:NetworkOk
c:
cd \
autoexec.bat
```

Create the file NFS\HOSTS, all fields starting with #@# will be set later when system is started by the TCP/IP BOOT-PROM:

```
127.0.0.1           localhost local
#@sip#########      #@t128######################
#@yip#########      #@chn######################
```

Create the file NFS\NETWORK.BAT which initializes the PC-NFS software:

```
net start rdr #@chn################## *
net route #@t128##################
net pcnfsd #@t128##################
net name *
net blip off
```

The file NFS\DRIVES.BAT will mount the network drive:

```
net use C: #@t128+###########:#@t130#################### /wsize=8192
```

The file ODI\NET.CFG loads the network driver and could look like:

```
Link Driver SMC8000
      Frame Ethernet_II
      Frame Ethernet_SNAP
      Frame Ethernet_802.2
      Frame Ethernet_802.3
```

Creating and installing a RAM disk image

The RAM disk image is a copy of the diskette just created.

Use the program BPSHELL to create that copy on MS-DOS systems. See "Creating RAM disk images" on page 55 for how to create RAM disk images. After a RAM disk image has been created, transfer it to the network server with ftp or copy the file to a mounted network drive. In the following procedure we assume that the RAM disk image is placed in /tftpboot/ramd.X[1] on the network server.

If you are working on a UNIX system the following command can be used:

```
# dd if=/dev/rfd0b of=/tftpboot/ramd.X bs=16k
# _
```

This assumes that /dev/rfd0b is your floppy drive, the argument bs=16k is only needed to speed up the floppy disk transfer and can be omitted.

Configuring the host files

The RAM disk image must be readable for all users. Check that the direc-

1. /tftpboot is a common place for diskless client files on UNIX systems. The extension .X indicates that the TCP/IP BOOT-PROM should load the RAM disk image into Extended Memory.

tory and the file have read permission. If not, you can set them by:

```
# chmod 755 /tftpboot
# ls -ld /tftpboot
drwxr-xr-x   2 root     other       32 Aug 15 13:32 /tftpboot
# chmod 744 /tftpboot/ramd.X
# ls -l /tftpboot/ramd.X
-rwxr--r--   1 root     other  1428480 Aug 15 13:33 /tftpboot/ramd.X
# _
```

After this, a user must be created on the network server and all relevant files copied into the directory of that user. These files include the complete MS-DOS and PC-NFS distribution as well as files needed for the user's application(s). Copy `COMMAND.COM` and `AUTOEXEC.BAT` to the home directory of the user. A possible file structure could look like:

```
# ls -l /usr/pcnfs
total 126
-rw-r--r--   1 pcnfs    msdos       1112 Aug 15 13:41 autoexec.bat
-rw-r--r--   1 pcnfs    msdos      52925 Aug 15 13:41 command.com
drwxr-xr-x   2 pcnfs    msdos       2608 Aug 15 13:42 dos
drwxr-xr-x   5 pcnfs    msdos       1344 Aug 15 13:42 nfs
drwxr-xr-x   2 pcnfs    msdos         64 Aug 15 13:43 tmp
drwxr-xr-x   4 pcnfs    msdos       2000 Aug 15 13:43 windows
# _
```

Edit the file `AUTOEXEC.BAT` to add some site specific commands:

```
@echo off

rem setup the search path for the host directories...
set PATH=c:\;c:\dos;c:\nfs
set COMSPEC=c:\command.com

rem this will free the TCP/IP BOOT-PROM memory space and drive A:...
a:\bin\bputil -r

rem usual AUTOEXEC.BAT stuff starts here...
prompt $p$g
set PATH=%PATH%;c:\windows
set TMP=c:\tmp
set TMPDIR=c:\tmp

rem set additional PC-NFS parameters...
set NFSDRIVE=c
set NFSPATH=c:\nfs

rem my private environment...
net use D: dksoft:/usr/bp/appl /wsize=8192
net use LPT2: lisa:Local_Printer /fmt=p /wsize=1024
net use LPT3: lisa:Color_Printer /fmt=p /wsize=1024
net use
```

```
loadhigh mouse
ver
```

Note - If MS-DOS files are edited on a UNIX system, check that all lines are terminated by CR/NL. The program UNIX2DOS converts files to be DOS conform.

Once again check if the user can access all files in his home directory. A simple way to set the owner of all files would be:

```
# find /usr/pcnfs -print | xargs chown pcnfs.msdos
# chown pcnfs.msdos /usr/pcnfs
# _
```

Configuring the BOOTP daemon

Now the BOOTP daemon configuration must be set up. Check with the chapter "Installing BOOTP and TFTP" on page 37 to see if there are any specific requirements for your operating system. Also check that the TFTP daemon is enabled.

A possible /etc/bootptab entry would be:

```
# template for vanilla Ethernet class C network
ether:\
      :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:


# template configuration for Sun PC-NFS
#     T128 is the NFS server's hostname
pcnfs:\
      :tc=ether:hd=/tftpboot:T128="dksoft":

# diskless client
#     T130 is the server directory to be mounted
diskless:\
      :tc=pcnfs:ha=0000c0595f66:ip=193.141.47.198:\
      :bf=ramd.X:\
      :T130="/usr/pcnfs":
```

Note - Some older and less functional BOOTP daemons have different formats for the /etc/bootptab file or a different place and name of their BOOTP daemon configuration file. Check your manual or use the BOOTP daemon on the TCP/IP BOOT-PROM *Utility Disk*.

This configuration uses two variables, `namely T128` which contains the name of your network server and `T130` which contains the user's home directory.

Note - Some BOOTP daemon cannot handle variables called tags. If you find that your BOOTP daemon cannot handle tags you must write its contents directly into the RAM disk image.

Testing the configuration

The configuration is now ready for use. If problems are encountered, check with the chapter "Troubleshooting and techniques" on page 137 and also with "Advanced techniques" on page 127 for more tips on how to configure diskless systems using the TCP/IP BOOT-PROM.

A sample PC/TCP installation

This chapter shows a sample installation using the PC/TCP and InterDrive network operating system. It is based on the following configuration:

- PC/TCP version 2.20
- MS-DOS 6.0
- SMC EtherCard Elite with Packet network driver
- A UNIX based network server

All the files shown in the following sample are also stored on the TCP/IP BOOT-PROM *Utility Diskette*.

Creating a bootable diskette

At first a bootable diskette must be created which holds the MS-DOS operating system, major PC/TCP programs to allow a network server logon and those programs which are started from `CONFIG.SYS`. All other programs are loaded from the network drive after a successful server logon.

Create a bootable MS-DOS diskette:

```
C:\> format a: /u/s
```

After this copy the MS-DOS and PC/TCP programs to your diskette. The files `BPUTIL.COM` and `BPUTIL.SYS` are from the TCP/IP BOOT-PROM *Utility Diskette*. The following table shows which files are needed on the diskette:

```
C:\> chkdsk a:\ /v
Volume PCTCP_SAMPL created 05-22-1993 8:27p
Volume Serial Number is 3D2F-1BE3
Directory A:\
A:\IO.SYS
A:\MSDOS.SYS
A:\COMMAND.COM
```

```
Directory A:\DOS
A:\DOS\EMM386.EXE
A:\DOS\HIMEM.SYS
A:\DOS\SUBST.EXE
Directory A:\PCTCP
A:\PCTCP\DRIVES.BAT
A:\PCTCP\IDRIVE.EXE
A:\PCTCP\IDRIVE.MSG
A:\PCTCP\PCTCP.INI
A:\PCTCP\IDMNT.EXE
A:\PCTCP\SMC_WD.COM
A:\PCTCP\ETHDRV.EXE
Directory A:\BIN
A:\BIN\BPUTIL.COM
A:\BIN\BPUTIL.SYS
Directory A:\ETC
A:\ETC\HOSTS
A:\CONFIG.SYS
A:\AUTOEXEC.BAT
C:\> _
```

Edit the file `CONFIG.SYS` to contain:

```
switches=/w
dos=high,umb

device=a:\bin\bputil.sys -f
device=a:\dos\himem.sys /testmem:off
device=a:\bin\bputil.sys -x
device=a:\dos\emm386.exe /x=d000-d3ff NOEMS /y=c:\dos\emm386.exe

buffers=5
files=40
shell=a:\command.com /p /e:1024

lastdrive=V
```

Edit the file `AUTOEXEC.BAT` to include the following lines:

```
rem setup the search path for the host directories...
set PATH=a:\;a:\dos;a:\pctcp
set COMSPEC=a:\command.com

rem make PC/TCP believe it's started from drive C:...
subst c: a:\

rem usual AUTOEXEC.BAT stuff starts here...
set TZ=MEZ0

rem set additional PC/TCP parameters...
```

```
set PCTCP=c:\pctcp\pctcp.ini

rem patch the BOOTP reply into PC/TCP configuration files...
a:\bin\bputil -a a:\etc\hosts
a:\bin\bputil -a a:\pctcp\drives.bat
a:\bin\bputil -a a:\pctcp\pctcp.ini

rem start PC/TCP...
loadhigh smc_wd 0x60 0x05 0x280 0xd000
loadhigh ethdrv
loadhigh idrive
subst c: /d
:NetworkLoop
set PCTCP=a:\pctcp\pctcp.ini
call drives
set PCTCP=c:\pctcp\pctcp.ini

rem test for network drive...
if exist c:\autoexec.bat goto NetworkOk
echo Failed to mount network drive !
goto NetworkLoop

rem switch over to network drive...
:NetworkOk
c:
cd \
autoexec.bat
```

Create the file `ETC\HOSTS`, all fields starting with `#@#` will be set later when system is started by the TCP/IP BOOT-PROM:

```
127.0.0.1           localhost local
#@sip#########      #@t128########################
#@yip#########      #@chn########################
```

The file `PCTCP\DRIVES.BAT` will mount the network drive:

```
rem do not use server hostname because /etc/hosts does not exist yet !
idmnt pctcp #@sip############## #@t130#################### c: pcnfs
```

The file `PCTCP\PCTCP.INI` configures PC/TCP and could look like:

```
[pctcp general]
host-name=#@chn####################
user=foobar
full-name=Joe Hacker
office-phone=12345679
time-zone=GMT
time-zone-offset=0
pfile=c:\etc\passwd
use-old-init-scheme=no
```

```
office=home

[pctcp kernel]
interface=ifcust 0
serial-number=0000-0000-0000
authentication-key=0000-0000-0000
window=2048
low-window=0
use-emm=yes
host-table=c:\etc\hosts
kernel-int=0x61
ip-precedence=routine
ip-precedence-matching=lax
ip-security=none
large-packets=8
small-packets=8
tcp-cons=8
udp-cons=8
tcp-connections=8
udp-connections=8
router-discovery=no


[pctcp ifcust 0]
subnet-mask=#@smf######################
ip-address=#@yip######################


[pctcp idrive]
cache-tmo=30
lock-tmo=3
map-char=~
retry-x=3
stream=2
sym-lnk=no
umask=775
automount=no
lock=no
mapping=RMAIL~00 RMAIL
share-tmo=0
default=
LongFileNames=0
```

Note - You must insert your serial number in order to start up the PC/TCP kernel.

Creating and installing a RAM disk image

The RAM disk image is a copy of the diskette you created.

Use the program `BPSHELL` to create that copy on MS-DOS systems. See "Creating RAM disk images" on page 55 for how to create RAM disk images. After you created a RAM disk image transfer it to the network server, you can use `ftp` or copy the file to a mounted network drive. In the following procedure we assume that the RAM disk image is placed in `/tftpboot/ramd.X`[1] on the network server.

If you are working on a UNIX system the following command will also do:

```
# dd if=/dev/rfd0b of=/tftpboot/ramd.X bs=16k
# _
```

This assumes that `/dev/rfd0b` is your floppy drive, the argument `bs=16k` is only needed to speed up the floppy disk transfer and can be omitted.

Configuring the host files

The RAM disk image must be readable for everyone. Check that the directory and the file have read permission. If not, you can set them by:

```
# chmod 755 /tftpboot
# ls -ld /tftpboot
drwxr-xr-x   2 root     other       32 Aug 15 13:32 /tftpboot
# chmod 744 /tftpboot/ramd.X
# ls -l /tftpboot/ramd.X
-rwxr--r--   1 root     other  1428480 Aug 15 13:33 /tftpboot/ramd.X
# _
```

1. /tftpboot is a common place for diskless client files on UNIX systems. The extension .X let's the TCP/IP BOOT-PROM load the RAM disk image into Extended Memory.

After this you must create a user on the network server and copy all files into the directory of that user. These files are the complete MS-DOS and PC/TCP distribution, there may also be files needed for the users application. Copy `COMMAND.COM` and `AUTOEXEC.BAT` to the home directory of the user. A possible file structure could look like:

```
# ls -l /usr/pctcp
total 126
-rw-r--r--   1 pctcp    msdos       1112 Aug 15 13:41 autoexec.bat
-rw-r--r--   1 pctcp    msdos      52925 Aug 15 13:41 command.com
drwxr-xr-x   2 pctcp    msdos       2608 Aug 15 13:42 dos
drwxr-xr-x   2 pctcp    msdos         64 Aug 15 13:42 etc
drwxr-xr-x   5 pctcp    msdos       1344 Aug 15 13:42 pctcp
drwxr-xr-x   2 pctcp    msdos         64 Aug 15 13:43 tmp
drwxr-xr-x   4 pctcp    msdos       2000 Aug 15 13:43 windows
# _
```

Edit file `AUTOEXEC.BAT`, you may want to add some site specific commands to it:

```
@echo off

rem setup the search path for the host directories...
set PATH=c:\;c:\dos;c:\pctcp
set COMSPEC=c:\command.com

rem this will free the TCP/IP BOOT-PROM memory space and drive A:...
a:\bin\bputil -r

rem usual AUTOEXEC.BAT stuff starts here...
prompt $p$g
set PATH=%PATH%;c:\windows
set TMP=c:\tmp
set TMPDIR=c:\tmp

rem set additional PC/TCP parameters...
set PCTCP=c:\pctcp\pctcp.ini
loadhigh predir config

rem my private environment...
idmnt appl dksoft /usr/bp/appl d: pcnfs msdos
loadhigh mouse
ver
```

Note - If you edit MS-DOS files on a UNIX system check that all lines are terminated by `CR/NL`. You can use the program `UNIX2DOS` to convert files.

Once again check if the user can access all files in his home directory. A simple way to set the owner of all files would be:

```
# find /usr/pctcp -print | xargs chown pctcp.msdos
# chown pctcp.msdos /usr/pctcp
# _
```

Configuring the BOOTP daemon

Now the BOOTP daemon configuration must be set up. You may want to check with chapter “Installing BOOTP and TFTP” on page 37 to see if there are any specifics to your operating system. Also check that the TFTP daemon is enabled.

A possible `/etc/bootptab` entry would be:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# template configuration for FTP-Software PC/TCP with InterDrive
#   T128 is the NFS server's hostname
pctcp:\
    :tc=ether:hd=/tftpboot:T128="dksoft":

# diskless client
#  T130 is the server directory to be mounted
diskless:\
    :tc=pctcp:ha=0000c0595f66:ip=193.141.47.198:\
    :bf=ramd.X:\
    :T130=”/usr/pctcp”:
```

Note - Some older and less functional BOOTP daemon have different formats of the `/etc/bootptab` file or a different place and name of their BOOTP daemon configuration file. Check your manual or use the BOOTP daemon on the TCP/IP BOOT-PROM *Utility Disk*.

This configuration uses two variables, `T128` which holds the name of your network server and `T130` which holds the user’s home directory.

Note - Some BOOTP daemon can not handle variables called tags. If you find that your BOOTP daemon can not handle tags you must write it’s contents directly into the RAM disk image.

Testing the configuration

The configuration is now ready for use. If you find problems check with the chapter “Troubleshooting and techniques” on page 137 you may also want to see “Advanced techniques” on page 127 for more tips how to configure diskless systems using the TCP/IP BOOT-PROM.

A sample TUN*TCP installation

This chapter shows a sample installation using the TUN*TCP network operating system. It is based on the following configuration:

- TUN*TCP version 7.10a
- MS-DOS 6.0
- SMC EtherCard Elite with Packet network driver
- A UNIX based network server

All the files shown in the following sample are also stored on the TCP/IP BOOT-PROM *Utility Diskette*.

Creating a bootable diskette

At first a bootable diskette must be created which holds the MS-DOS operating system, major TUN*TCP programs to allow a network server logon and those programs which are started from `CONFIG.SYS`. All other programs are loaded from the network drive after a successful server logon.

Create a bootable MS-DOS diskette:

```
C:\> format a: /u/s
```

After this copy the MS-DOS and TUN*TCP programs to your diskette. The files `BPUTIL.COM` and `BPUTIL.SYS` are from the TCP/IP BOOT-PROM *Utility Diskette*. The following table shows which files are needed on the diskette:

```
C:\> chkdsk a:\ /v
Volume TUN_SAMPL created 05-22-1993 8:27p
Volume Serial Number is 2D2B-15ED
Directory A:\
A:\IO.SYS
A:\MSDOS.SYS
A:\COMMAND.COM
```

```
A:\AUTOEXEC.BAT
A:\CONFIG.SYS
Directory A:\TUNTCP
A:\TUNTCP\ETHTCP.EXE
A:\TUNTCP\SMC_WD.COM
A:\TUNTCP\TUNNFS.EXE
A:\TUNTCP\TELL.EXE
A:\TUNTCP\ESTCPIF.SYS
A:\TUNTCP\ESTCPIP.SYS
A:\TUNTCP\IFCONF.EXE
A:\TUNTCP\IPCONF.EXE
A:\TUNTCP\MOUNT.EXE
Directory A:\ETC
A:\ETC\HOSTS
Directory A:\BIN
A:\BIN\BPUTIL.COM
A:\BIN\BPUTIL.SYS
Directory A:\DOS
A:\DOS\EMM386.EXE
A:\DOS\HIMEM.SYS
A:\DOS\SUBST.EXE
C:\> _
```

Edit the file `CONFIG.SYS` to contain:

```
switches=/w
dos=high,umb

device=a:\bin\bputil.sys -f
device=a:\dos\himem.sys /testmem:off
device=a:\bin\bputil.sys -x
device=a:\dos\emm386.exe /x=d000-d3ff NOEMS /y=c:\dos\emm386.exe

device=a:\bin\bputil.sys -a a:\etc\hosts
device=a:\bin\bputil.sys -a a:\autoexec.bat

devicehigh=a:\tuntcp\estcpif.sys
devicehigh=a:\tuntcp\estcpip.sys

buffers=5
files=40
shell=a:\command.com /p /e:1024

lastdrive=V
```

Edit the file `AUTOEXEC.BAT` to include the following lines:

```
@echo off

rem setup the search path for the host directories...
```

```
set PATH=a:\;a:\dos;a:\tuntcp
set COMSPEC=a:\command.com

rem make TUN PLUS believe it's started from drive c:...
subst c: a:\
c:

rem configure TUN PLUS...
ifconf ifcust0 /ip #@yip############ /submask #@smf############
ipconf $ipcust /hostname #@chn############ /hosttable c:\etc\hosts

rem start drivers and TUN PLUS...
loadhigh smc_wd 0x60 0x05 0x280 0xd000
loadhigh ethtcp.exe -t 8 -u 3 -p 12
loadhigh tunnfs.exe

rem mount network drive...
a:
subst c: /d

rem can not use hostname here because c:\etc\hosts does not exist yet !
:NetworkLoop
mount #@t128####### #@sip########### #@t129############# c: user

rem test for network drive...
if exist c:\autoexec.bat goto NetworkOk
echo Failed to mount network drive !
goto NetworkLoop

rem switch over to network drive...
:NetworkOk
c:
cd \
autoexec.bat
```

Create the file `ETC\HOSTS`, all fields starting with `#@#` will be set later when system is started by the TCP/IP BOOT-PROM:

```
#@sip############ #@t128#####################
#@yip############ #@chn######################
```

Creating and installing a RAM disk image

The RAM disk image is a copy of the diskette you created.

Use the program `BPSHELL` to create that copy on MS-DOS systems. See “Creating RAM disk images” on page 55 for how to create RAM disk images. After you created a RAM disk image transfer it to the network server, you can use ftp or copy the file to a mounted network drive. In the following procedure we assume that the RAM disk image is placed in /tftpboot/ramd.X[1] on the network server.

If you are working on a UNIX system the following command will also do:

```
# dd if=/dev/rfd0b of=/tftpboot/ramd.X bs=16k
# _
```

This assumes that /dev/rfd0b is your floppy drive, the argument bs=16k is only needed to speed up the floppy disk transfer and can be omitted.

Configuring the host files

The RAM disk image must be readable for everyone. Check that the directory and the file have read permission. If not, you can set them by:

```
# chmod 755 /tftpboot
# ls -ld /tftpboot
drwxr-xr-x   2 root     other       32 Aug 15 13:32 /tftpboot
# chmod 744 /tftpboot/ramd.X
# ls -l /tftpboot/ramd.X
-rwxr--r--   1 root     other  1428480 Aug 15 13:33 /tftpboot/ramd.X
# _
```

After this you must create a user on the network server and copy all files into the directory of that user. These files are the complete MS-DOS and TUN*PLUS distribution, there may also be files needed for the users application. Copy COMMAND.COM and AUTOEXEC.BAT to the home directory of the user. A possible file structure could look like:

```
# ls -l /usr/tuntcp
total 126
-rw-r--r--   1 tuntcp    msdos       1112 Aug 15 13:41 autoexec.bat
-rw-r--r--   1 tuntcp    msdos      52925 Aug 15 13:41 command.com
drwxr-xr-x   2 tuntcp    msdos       2608 Aug 15 13:42 dos
drwxr-xr-x   2 tuntcp    msdos         64 Aug 15 13:42 etc
drwxr-xr-x   5 tuntcp    msdos       1344 Aug 15 13:42 tuntcp
drwxr-xr-x   2 tuntcp    msdos         64 Aug 15 13:43 tmp
drwxr-xr-x   4 tuntcp    msdos       2000 Aug 15 13:43 windows
# _
```

Edit file AUTOEXEC.BAT, you may want to add some site specific commands to it:

```
@echo off
```

1. /tftpboot is a common place for diskless client files on UNIX systems. The extension .X let’s the TCP/IP BOOT-PROM load the RAM disk image into Extended Memory.

```
rem setup the search path for the host directories...
set PATH=c:\;c:\dos;c:\tuntcp
set COMSPEC=c:\command.com

rem this will free the TCP/IP BOOT-PROM memory space and diskette drive
A:...
a:\bin\bputil -r

rem usual AUTOEXEC.BAT stuff starts here...
prompt $p$g
set PATH=%PATH%;d:\windows
set TMP=c:\tmp
set TMPDIR=c:\tmp

rem my private environment...
mount appl 193.141.47.193 /usr/bp/appl d: user msdos
loadhigh keyb gr
loadhigh mouse
ver
```

Note - If you edit MS-DOS files on a UNIX system check that all lines are terminated by `CR/NL`. You can use the program `UNIX2DOS` to convert files.

Once again check if the user can access all files in his home directory. A simple way to set the owner of all files would be:

```
# find /usr/tuntcp -print | xargs chown tuntcp.msdos
# chown tuntcp.msdos /usr/tuntcp
# _
```

Configuring the BOOTP daemon

Now the BOOTP daemon configuration must be set up. You may want to check with chapter “Installing BOOTP and TFTP” on page 37 to see if there are any specifics to your operating system. Also check that the TFTP daemon is enabled.

A possible `/etc/bootptab` entry would be:

```
# template for vanilla Ethernet class C network
ether:\
    :hn:sm=255.255.255.0:vm=rfc1048:ht=ethernet:

# template configuration for Esker TUN*PLUS
#   T128 is the NFS server's hostname
tuntcp:\
    :tc=ether:hd=/tftpboot:T128="dksoft":
```

```
# diskless client
#  T129 is the server directory to be mounted
diskless:\
    :tc=tuntcp:ha=0000c0595f66:ip=193.141.47.198:\
    :bf=ramd.X:\
    :T129="/usr/tuntcp":
```

Note - Some older and less functional BOOTP daemon have different formats of the `/etc/bootptab` file or a different place and name of their BOOTP daemon configuration file. Check your manual or use the BOOTP daemon on the TCP/IP BOOT-PROM *Utility Disk*.

This configuration uses two variables, `T128` which holds the name of your network server and `T129` which holds the user's home directory.

Note - Some BOOTP daemon can not handle variables called tags. If you find that your BOOTP daemon can not handle tags you must write it's contents directly into the RAM disk image.

Testing the configuration

The configuration is now ready for use. If you find problems check with the chapter "Troubleshooting and techniques" on page 137 you may also want to see "Advanced techniques" on page 127 for more tips how to configure diskless systems using the TCP/IP BOOT-PROM.

Troubleshooting and techniques

Troubleshooting the TCP/IP BOOT-PROM

The system hangs while downloading a RAM disk image

Problem: The PC downloads a RAM disk image and hangs somewhere between 0x200 and 0x260 K byte.

Solution: The filename of the RAM disk image does not end with the capital letter `X`. Therefore the TCP/IP BOOT-PROM loads the RAM disk image into Base Memory and overwrites internal structures. Add `X` to the filename of the RAM disk image to make the PROM upload the image into Extended Memory, e.g. `/tftpboot/ramd.X`.

The PROM aborts with ‚M30: can't ARP tftp address'

Problem: The TCP/IP BOOT-PROM tried to get the TFTP servers MAC-address using the ARP protocol but the TFTP server did not reply.

Solution: Check if the network path to the TFTP server is set up correctly. If you use a router, check if it is configured to allow access to the TFTP server. The address to be used for the TFTP server is given by the BOOTP reply. If the BOOTP server fills out the RFC-951 BOOTP protocol gateway field, the PROM uses this entry as its gateway to the TFTP server.

The PROM aborts with ‚M32: TFTP open time-out occurred'

Problem: The PROM does not get an acknowledge from the TFTP server.

Solution: Check if the TFTP daemon is enabled in `/etc/inetd.conf`. Some TFTP daemons do not answer TFTP open requests if the file is not accessible for all users or the file does not exist. Also check if the path to the file is accessible.

The PROM aborts with ‚M41: BOOTP time-out occurred'

Problem: The PROM did not get an answer from a BOOTP server.

Solution: Check if your BOOTP daemon is running, if an entry for the client is made, if there are errors in the BOOTP configuration file (e.g. `/etc/bootptab`). You may also check with your manual if a debug mode for the BOOTP daemon is available. If you use a router between the client and the BOOTP server check if the router is able to route IP broadcasts. Many routers use IP helper addresses to do so.

The PROM aborts with ‚M42: no client or server IP'

Problem: The client or server IP address field of the BOOTP reply is not set by the BOOTP server.

Solution: Check if the BOOTP daemon configuration (e.g. `/etc/bootptab`) is set up correctly.

The PROM aborts with ‚M43: no bootfilename'

Problem: The bootfilename field of the BOOTP reply is not set.

Solution: There is a valid entry for the client in the BOOTP configuration file but the bootfilename does not exist or is not accessible. Also check if the path to the file is accessible to all users.

The PROM aborts with ‚E6E: can not start from RAM disk'

Problem: The PROM did not find the bootable ID in the bootsector of the RAM disk.

Solution: Check if the diskette from which the RAM disk image was made is bootable. Verify that the RAM disk image was transferred binary, e.g. FTP will strip off the 8th bit if ASCII mode is enabled. If you use DOS versions earlier than 4.0 which did not have the bootable ID make a 18 sector RAM disk image.

The PROM aborts with error ‚T??: access violation'

Problem: The TFTP-daemon can not access the RAM disk image.

Solution: Check that all directories and the file itself have read permission to all users. Check if the TFTP daemon is restricted to a directory via the -s option and the BOOTP daemon is not restricted to the same directory.

Programs in High Memory are no longer accessible

Problem: Using EMM386 with the RAM or NOEMS options causes that the configuration of the High Memory (memory between 640K and 1M byte) gets lost after the BPUTIL program has restored the memory allocated by the RAM disk.

It shows that it is not possible to use MEM /C or LOADHIGH after the BPUTIL -R command has been executed. Programs which are already loaded into High Memory are not influenced by this problem.

Solution: Execute BPUTIL -F before you load HIMEM.SYS and EMM386.EXE in CONFIG.SYS:

```
device=bputil.sys -f
device=himem.sys /testmem:off
device=bputil.sys -x
device=emm386.exe noems
```

EMM386 hangs with DMA based network controllers

Problem: On some systems EMM386 claims that the number of DMA handles are exceeded and halts the system. This only appears when the RAM disk image is loaded into Extended Memory.

Solution: Increase the number of DMA handles via the /D=nnn option of EMM386. If this does not help, check with the manufacturer of your network controllers device-driver. This problem is not caused by the TCP/IP BOOT-PROM. It is caused by the memory transfer between Extended and Base Memory. Some DMA based network drivers can not handle this.

Using EMM386 on a diskless system

Problem: When starting MS-Windows it can not find EMM386.

Solution: Include the /Y option if EMM386 is used on a diskless system. This helps programs that switch into Protected Mode to find EMM386.EXE since it has been started from the RAM disk drive which does no longer exist after BPUTIL -R is executed.
A sample CONFIG.SYS file entry could look like:

```
device=emm386.exe /y=c:\dos\emm386.exe
```

Using MS-Windows 3.0 on a diskless system

Problem: Running MS-DOS with Microsoft Windows 3.0 requests to specify the location of the `WINA20.386` file.

Solution: This can be done by adding the line:

```
switches=/w
```

To your `CONFIG.SYS` file. Also you must include the statement:

```
[386Enh]
device=c:\dos\wina20.386
```

Into the `SYSTEM.INI` file of MS-Windows. See the MS-DOS manual for more information.

Formatting the boot drive from MS-Windows

Problem: If the PC is booted using the TCP/IP BOOT-PROM, the MS-Windows diskette format menu does not work on the boot drive, e.g. diskette drive `A:`.

Solution: Since MS-Windows disables formatting of the boot drive use the MS-DOS `FORMAT` program to format a diskette in drive `A:`.

Using the BPUTIL patching options

Problem: The RFC1048 protocol limits the amount of user tags (additional information) on the BOOTP reply to 64 bytes.
If you find that information you have included in your */etc/bootptab* file is not transferred to your diskless system, this is caused by the limitation of the protocol.

Solution: Do not use tags directly to set your data. You can save tag space by using a tag to point to a file (e.g. MS-DOS batch file) and then execute this file to insert your data.

Booting over a router

Problem: To find its network address and configuration, the TCP/IP BOOT-PROM uses the BOOTP protocol. This BOOTP request must be done by broadcasting over the network. Since routers do not route MAC (e.g. network hardware address) broadcasts it is not possible to bootstrap a PC from a server which is located in a different network.

Solution: Some routers are able to do MAC routing or use IP helper addresses to handle BOOTP broadcasts. Otherwise bridges must be used in order to transfer broadcasts between two networks.

Advanced Techniques

Using the memory occupied by the ROM

You can use the memory area which is occupied by the program code in the ROM for uploading programs. Use `EMM386` to do so:

```
device=emm386.exe i=cc00-cfff noems
```

This includes the memory range `0xcc00` to `0xcfff` which is allocated by the ROM on the network adapter for uploading MS-DOS programs. The size and location of the ROM area depends on the network card and configuration.

Faster TFTP transfer

The TCP/IP BOOT-PROM is able to adapt to non RFC standard TFTP packet sizes. If the TFTP server sends a larger packet than the RFC standard of 512 bytes, the TCP/IP BOOT-PROM will also work with the extended packet size.

Using larger packet sizes decreases the number of transmitted packets, which results in shorter boot time and reduced server load.

Note - Packet sizes larger than 512 bytes will not work together with application which can not adapt to the packet size or reply on the standard RFC specification.

In order to use standard RFC (512 bytes) and non RFC standard TFTP servers in one network concurrently you can bind the non RFC standard TFTP daemon to UDP port 59 (reserved for private file transfer service). The standard RFC port for the TFTP daemon is 69 (reserved for TFTP file transfer service).

If the bootfilename is tagged by the capital letter `P` (e.g. `/tftpboot/ramd.X`) or `PX` if the image is to be loaded in Extended Memory (e.g. `/tftpboot/ramd.PX`), the TCP/IP BOOT-PROM first tries to connect to the TFTP daemon at port 59. If the TFTP request is not answered after 3 tries, the TCP/IP BOOT-PROM falls back to the standard RFC port 69.

Using larger RAM disk images

If the RAM disk can not hold all the data you need to start your operating system and network software you can delete already loaded programs from the RAM disk and transfer new files using the `BPUTIL -T` option.

You may also want to create a larger RAM disk using the MS-DOS `RAMDRV` program and extract a compressed archive to this RAM disk.

Tables

Supported network controllers

The following table shows the supported network controllers, their TCP/IP BOOT-PROM network driver names and options.

The ‘Supported settings’ column names all supported settings, they may be changed using the `BPSHELL` program. The ‘Tested settings’ column shows a sample configuration on which the TCP/IP BOOT-PROM has been tested.

Drivername Controller-name	**Drv #**	**Features**													**Supported settings**			**Tested settings**			
		8K ROM[a]	16K ROM[b]	32 K ROM[c]	64K ROM[d]	Long diagnostic[e]	Verbose option[f]	MC-POS[g]	Attached DMA[h]	DHCP[i]	EISA[j]	PCI[k]	PCMCIA[l]	MTFTP[m]	I/O	Memory	DMA	I/O	ROM	Memory	DMA
AT1700 Allied Telesis AT-1700	0028		✔			✔	✔			✔				✔	300 280 2a0 240 300			280	cc00		
C503 Etherlink II Etherlink II/16	0001	✔													Detect			300		cc00	
C505 Etherlink Plus	0016	✔													300 280 310 330 360			300	cc00		

Drivername Controller-name	Drv #	Features													Supported settings			Tested settings			
		8K ROM[a]	16K ROM[b]	32 K ROM[c]	64K ROM[d]	Long diagnostic[e]	Verbose option[f]	MC-POS[g]	Attached DMA[h]	DHCP[i]	EISA[j]	PCI[k]	PCMCIA[l]	MTFTP[m]	I/O	Memory	DMA	I/O	ROM	Memory	DMA
C507 Etherlink 16	0014		✔			✔	✔			✔					300 280 310 330 360	De- tect		300	cc00	d000 16K	
C509 Etherlink III 3C509, 3C509B, 3C579	0017		✔			✔	✔			✔	✔			✔	De- tect			300	cc00		
C523 Etherlink/MC	0011	✔						✔							Detect			300		c800	
C589 Etherlink III 3C589, 3C589B	0030		✔			✔	✔			✔			✔	✔	De- tect	De- tect		3E8/ 300	cc00	d000	
C590 Etherlink III 3C590	002E		✔			✔	✔			✔		✔		✔	De- tect			540			
C595 Etherlink III 3C595	002F		✔			✔	✔			✔		✔		✔	De- tect			540			
C597 Etherlink III 3C597	0031		✔			✔	✔			✔	✔			✔	De- tect			4000			
CTRON21 E11xx E21xx	0020				✔	✔	✔			✔				✔	380 280 300 200	d000 e000 c000		380	d000	d000	
E118 Eltec E118	0024		✔			✔	✔			✔				✔	320 360		5 5	320			5
EBLAST INTERLAN EtherBlaster	0023		✔			✔	✔			✔				✔	300 320 340 360		5 3 6 7	300			5

Drivername Controller-name	Drv #	Features													Supported settings			Tested settings			
		8K ROM[a]	16K ROM[b]	32 K ROM[c]	64K ROM[d]	Long diagnostic[e]	Verbose option[f]	MC-POS[g]	Attached DMA[h]	DHCP[i]	EISA[j]	PCI[k]	PCMCIA[l]	MTFTP[m]	I/O	Memory	DMA	I/O	ROM	Memory	DMA
ED586 EDEN ED586/8	0002		✔			✔	✔			✔					3e8 2c8 2e8 3c8	de00 d600 ce00 c600		3e8	cc00	de00	
ETH16I ICL EtherTeam 16i and C4	0019		✔			✔	✔			✔				✔	2a0 240 300 280 340			2a0	cc00		
EWORKS2 EtherWORKS LC and Turbo	0015		✔			✔	✔			✔				✔	300 200	De- tect		300	dc00	d000	
EWORKS3 EtherWORKS DE203, DE204, DE205	001b		✔			✔	✔			✔					300 200 280 240 2a0			300	cc00		
EXOS105 EXOS 105 EXOS 105- 10T	000f		✔			✔	✔		✔	✔				✔	300 320 340 360		5 6 7 3	340	c800		7
EXOS205 EXOS 205	0013		✔			✔	✔			✔					310 300 320 330	d000 da00 cc00 d800 d400		310	cc00	d000	
EXP16 EtherExpress 16	000b		✔			✔	✔			✔					300 240 310 330 360			300	c800		
EXP16F EtherExpress 16 FLASH	001e		✔			✔	✔			✔					300 240 310 330 360			300	c800		

Drivername Controller-name	Drv #	Features													Supported settings			Tested settings			
		8K ROM[a]	16K ROM[b]	32 K ROM[c]	64K ROM[d]	Long diagnostic[e]	Verbose option[f]	MC-POS[g]	Attached DMA[h]	DHCP[i]	EISA[j]	PCI[k]	PCMCIA[l]	MTFTP[m]	I/O	Memory	DMA	I/O	ROM	Memory	DMA
HPWCE Hewlett Pack-ard Windows Client Ether-net	0021			✔		✔	✔		✔	✔				✔	300 320 340 360		5 6 7 3	300			5
HPWCT Hewlett Pack-ard Windows Client Token Ring	0027			✔		✔	✔			✔				✔	a20 a00 a40 a60 a80		Detect	a20			3
IBMAETH IBM LAN Adapter/A	002b		✔			✔	✔	✔		✔				✔	Detect			280	cc00	d800	
ISO4102 ISOLAN 4102	0003		✔			✔	✔		✔	✔				✔	280 250 390 330 360		0 1 3 5 6	280	cc00		0
NE1000 Novell NE1000	000d	✔													300 320 340 360			300	c800		
NE2000 Novell NE2000	000e		✔			✔	✔			✔				✔	300 320 340 360			300	c800		
NE2000P Novell NE2000 Plus	001f	✔													300 320 340 360			300	c800		
NE2100 Novell NE2100	0010		✔			✔	✔		✔	✔				✔	300 320 340 360		5 6 7 3	300	c800		5

Drivername Controller-name	Drv #	Features													Supported settings			Tested settings			
		8K ROM[a]	16K ROM[b]	32 K ROM[c]	64K ROM[d]	Long diagnostic[e]	Verbose option[f]	MC-POS[g]	Attached DMA[h]	DHCP[i]	EISA[j]	PCI[k]	PCMCIA[l]	MTFTP[m]	I/O	Memory	DMA	I/O	ROM	Memory	DMA
NI5210 INTERLAN NI5210	0004	✔													360 320 300 340 280	cc00 d000 d400 d800 dc00		360	cc00		
NI6510 INTERLAN NI6510	0005		✔			✔	✔			✔				✔	320 360 300 340		Detect	360	dc00		3
NI9210 INTERLAN NI9210	0006	✔						✔							Detect			300	d400	d000 16K	
PCNET Microdyne NE5500	002c		✔			✔	✔			✔		✔		✔	De-tect			540			
PKTDRV Crynwr Packet Drivers	001c		✔		✔	✔				✔					60 61 62 63 64	Detect		SMC_WD Packet Driver			
PD8023 Pure Data 8023	001a	✔													280 240 300 260 360	De-tect		280	cc00	d000	
SKJUN SK-NET jun-ior	000c		✔			✔	✔			✔				✔		c400 cc00 c800 d000 dc00			c800	cc00	
SKG16 SK-NET G8 SK-NET G16	0007		✔			✔	✔			✔				✔	390 328 288 220 320	pla-ced be-hind ROM		390	cc00	d000	

Drivername Controller-name	Drv #	Features													Supported settings			Tested settings			
		8K ROM[a]	16K ROM[b]	32 K ROM[c]	64K ROM[d]	Long diagnostic[e]	Verbose option[f]	MC-POS[g]	Attached DMA[h]	DHCP[i]	EISA[j]	PCI[k]	PCMCIA[l]	MTFTP[m]	I/O	Memory	DMA	I/O	ROM	Memory	DMA
SKMC2 SK-NET MC2	0008		✔			✔	✔	✔		✔				✔	Detect			390	cc00	d000	
SKYNET SK-Y-NET	0012		✔			✔	✔		✔	✔				✔	300 320 340 360		5 6 7 3	360	cc00		3
SMC8115 SMC Token-Card SMC8115	0018				✔	✔	✔			✔					Detect			280	d000	cc00	
SMC8232 SMC Ether-Card Elite32 SMC8232	0025			✔		✔	✔			✔	✔			✔	Detect			e400	d800	d000	
SMC8332 SMC Token-Card Elite32 SMC 8332	0026				✔	✔	✔			✔	✔				Detect			e400	d000	cc00	
SMC8416 SMC Ether-Card Elite/Ultra 8013, 8216, 8416	001d			✔		✔	✔			✔				✔	Detect			280	d000	cc00	
SMC8432 SMC Ether-Power SMC 8432 DEC21041	002d				✔	✔	✔			✔		✔		✔	Detect			540			
WD8ISA WD8003EB WD8003EP WD8013EBT WD8013EP SMC8013EPC	0009		✔			✔	✔			✔				✔	240 280 300 260 360	d000 da00 cc00 d800 d400		280	cc00 c800	d000	

Drivername Controller-name	Drv #	Features													Supported settings			Tested settings			
		8K ROM[a]	16K ROM[b]	32 K ROM[c]	64K ROM[d]	Long diagnostic[e]	Verbose option[f]	MC-POS[g]	Attached DMA[h]	DHCP[i]	EISA[j]	PCI[k]	PCMCIA[l]	MTFTP[m]	I/O	Memory	DMA	I/O	ROM	Memory	DMA
WD8MCA	000a		✔			✔	✔	✔		✔				✔	Detect						
WD8003/A																		280	cc00	d000	
SMC8013EP/A																		800	cc00	d000	

a. 8K ROMs hold 8192 byte of program code, they are 2764 EPROM type.
b. 16K ROMs hold 16384 byte of program code, they are 27128 EPROM type.
c. 32K ROMs hold 32768 byte of program code, they are 27256 EPROM type.
d. 64K ROMs hold 65536 byte of program code, they are 27512 EPROM type.
e. By pressing the key ‘d’ while the ROM starts up, the long diagnostic can be activated. This shows the configuration and traces the network activity.
f. By pressing the key ‘v’ while the ROM starts up, the verbose option can be activated. This gives information about the ROMs activity.
g. Microchannel controller detect their configuration via the POS registers which are set using the ‘PS/2 Reference Disk’.
h. On some controllers the DMA configuration can not be detected via software. On these controllers the DMA channel depends on the configured i/o port.
i. This controller supports the DHCP protocol in addition to the BOOTP protocol.
j. This controller implements EISA auto detection and configuration.
k. This controller implements PCI auto detection and configuration.
l. This controller implements PCMCIA auto detection and configuration.
m. This controller supports multicast TFTP in addition to the TFTP protocol.

Error messages

Some Ethernet controller have only a small amount of ROM space available (e.g. controllers with a 8 K byte ROM socket). On these controllers the verbose error messages are replaced by their numeric code.

You can look up the error message text using the following table:

code	error message text
M10	`ARP canceled by keystroke` The ARP request has been canceled because the user pressed a key.
M11	`ARP time-out occurred` The ARP request has not been answered.
M20	`can't copy memory` The System BIOS (int 15h, service 87h) was not able to copy data.
M21	`can't write to memory` There is not enough memory to store all MTFTP packages.
M22	`can't write to memory` The RAM memory is to low and reached the ROM range.
M30	`can't ARP tftp address` The TFTP server did not answer the ARP request.
M31	`TFTP open canceled by keystroke` The TFTP open has been canceled because the user pressed a key.
M32	`TFTP open time-out occurred` The TFTP server did not answer or the TFTP daemon is disabled.
M33	`unknown TFTP opcode` The TFTP server sends unknown packets.
M34	`TFTP read canceled by keystroke` The TFTP read has been canceled because the user pressed a key.
M35	`TFTP read time-out occurred` The TFTP server did not answer even that the open request was granted.
M36	`TFTP write canceled by keystroke` The TFTP write has been canceled because the user pressed a key.
M37	`TFTP write time-out occurred` The TFTP server did not answer even that the open request was granted.
M38	`can't open TFTP connection` The TFTP server did not answer or is disabled.
M39	`can't read from TFTP connection` The TFTP server did not answer even that the open request was granted.

M40	`BOOTP canceled by keystroke`
	`DHCP canceled by keystroke`
	The DHCP/BOOTP request has been canceled because the user pressed a key.

code	error message text

M41	`BOOTP time-out occurred`
	`DHCP time-out occurred`
	The DHCP/BOOTP server did not answer, is disabled or has no entry for this client.

M42	`no client or server IP` The BOOTP request was answered but did not contain a client or server IP.
M43	`no bootfilename` The BOOTP request was answered but did not contain a filename because the file does not exist or is unreadable.
M50	`FFRAME canceled by keystroke` The RPL FIND FRAME command has been canceled because the user pressed a key.
M51	`FFRAME timeout occurred` The RPL FIND FRAME command has not been answered.
M52	`SFREQUEST canceled by keystroke` The RPL SEND FILE REQUEST command has been canceled because the user pressed a key.
M53	`SFREQUEST time-out occurred` The RPL SEND FILE REQUEST command was not answered even that the FILE FRAME command was granted.

M6f	`System is locked!`
	`Press <Ctrl>-<Alt>-<Del> to reboot.`
	The diskless boot failed and the system is now locked.

M70	`canceled by keystroke` The checksum procedure has been canceled because the user pressed a key.
M80	`RARP canceled by keystroke` The RARP request has been canceled because the user pressed a key.
M81	`RARP time-out occurred` The RARP request has not been answered.
M82	`no client IP` The RARP request has been answered but did not contain a client IP.
M90	`can't init controller for multicast` Failed to re-initialize the network controller to receive multicast packages.

code	error message text
M91	`MTFTP open canceled by keystroke` The MTFTP open has been canceled because the user pressed a key.
M92	`MTFTP open time-out occurred` The MTFTP server did not answer or the TFTP daemon is disabled.
M93	`unknown MTFTP opcode` The MTFTP server sends unknown packets.
M94	`MTFTP read canceled by keystroke` The MTFTP read has been canceled because the user pressed a key.
M95	`MTFTP read time-out occurred` The MTFTP server did not answer even that the open request was granted.
M96	`can't ARP mtftp address` The MTFTP server did not answer the ARP request.
M98	`can't open MTFTP connection` The MTFTP server did not answer or is disabled.
M99	`can't read from MTFTP connection` The MTFTP server did not answer even that the open request was granted.
M9a	`too many MTFTP packages` The MTFTP server is sending more packages as can be stored.
E60	`ROM disabled via setup` The network controller setup disables the ROM.
E61	`Keystroke BOOT-PROM aborted` The TCP/IP BOOT-PROM has been aborted because the user pressed a key.
E62	`can't init controller (???)` The network controller can not be initialized.
E63	`can't init controller (i/o port)` The network controller can not be found.
E64	`can't init controller (shm)` The network controller's shared memory can not be set.
E65	`can't init shm (using 0xd0000)` The network controller's shared memory is forced to `0xd0000`.
E66	`can't init controller (shm)` The network controller's shared memory can not be set.
E67	`can't init controller (???)` The network controller can not be initialized.

code	error message text
E68	`can't init controller (MC-ID)` The network controller can not be found in any MCA slot.
E69	`can't init controller (shm)` The network controller's shared memory can not be set.
E6a	`ROM not licensed to this controller` The network controller's MAC address is not the same as the one the ROM is licensed to.
E6b	`can not find RPL server` The RPL server did not answer.
E6c	`can not find RARP server` The RARP server did not answer.
E6d	`can not find BOOTP server` The BOOTP server did not answer.
E6e	`can not start from RAM disk image` The RAM disk image does not contain a valid bootsector.
E71	`invalid checksum` The RAM disk image does not have a valid checksum.
E72	`unknown boot protocol` No boot protocol (DHCP/BOOTP/RARP/RPL) has been configured.
T??	`<TFTPD error string>` A TFTP error occurred, `??` holds the TFTP error code.

Index

U

W

X

TCP/IP BOOT-PROM
User's Manual